R-Space X-ray Absorption Package (RSXAP)

Corwin Booth
Lawrence Berkeley National Laboratory
MS 70A-1150
Berkeley, CA 94720
(510)486-6079
chbooth@lbl.gov

Table of Contents

I. Introduction

A. Overview

RSXAP is a collection of programs for reducing and fitting x-ray-absorption fine-structure (XAFS) data. The authors are strong proponents of fitting in r-space, and as such, the codes are strongly focused on reducing the raw spectra to produce reasonable looking r-space data.

The primary programs are SUREDUCE and RSFIT. Their main features include:

- Robust filtering procedures that allow fitting over desired regions in both k- and r-space

- A variety of methods for determining mu_0 backgrounds, including fitting the low-r amplitude of a spectra to a choice of functions, and employing the so-called "iterative technique" for analyzing fit-residuals in e-space

- An improved method for self-absorption correction of fluorescence data (correction is k-dependent, currently unpublished).

- Cumulant analysis to determine the third and fourth moments of a given atomic shells radial distribution function.

- The ability to quickly and easily view data while in the reduction process.

The package also has utilities for fitting XANES data and many utilities for shifting energies, averaging files, etc. etc.

For more information about the theory behind these codes and their applicability to real-world systems, please consult the following references:

T. M. Hayes and J. B. Boyce, in *Solid State Physics*, ed. by H. Ehrenreich, F. Seitz, and D. Turnbull (Academic, New York, 1982), Vol. 37, p. 173.

G. G. Li, F. Bridges, and C. H. Booth, Phys. Rev. B 52, 6332 (1995).

F. Bridges, C. H. Booth, and G. G. Li, Physica B 208&209, 121 (1995).

C. H. Booth, Ph.D. Thesis, University of California, Santa Cruz, 1996.

For more information and the latest codes, please see http://lise.lbl.gov/RSXAP.

B. The Manual

This manual is very much a work in progress, and is really at the beginning stages. Consequently, it is incomplete and probably opaque. That said, I am trying to stick to some conventions. Text as it appears on the screen is in `courier 10pt`. When a piece of such text includes user responses, they will appear as **`bold`**. All program names are to be typed in lower case, but in this manual they will be referred to in UPPER CASE.

Sometimes examples will require the use of a terminal. The terminal prompt is represented by a ">" on the far left of the page. Often, hitting the ENTER key is required after a command is typed. Sometimes this will be explicit when I refer to a "<RET>". Some programs require user input in the form of file names or directory names. These are represented in the examples as **<file name>**, **<directory name>**, etc.

Please also note that the codes are "living" and some menu options may change. For the most part, please be aware that the most common change will be that an option *moves*, so keep your eyes open!

Finally, this manual is NOT an introduction to the XAFS technique! There is a little bit of that in here, but you should have a good book (Say, B. K. Teo, *EXAFS: Basic Principles and Data Analysis* (Spinger-Verlag, New York)) and a good guru before embarking on this adventure.

II. Summary and Quick Start

Assuming RSXAP is already installed on your system, here's what you need to know!

A. The Environment

Before you can really use these codes, certain environment variables have to be set. A sample .bash_profile file should look like this:

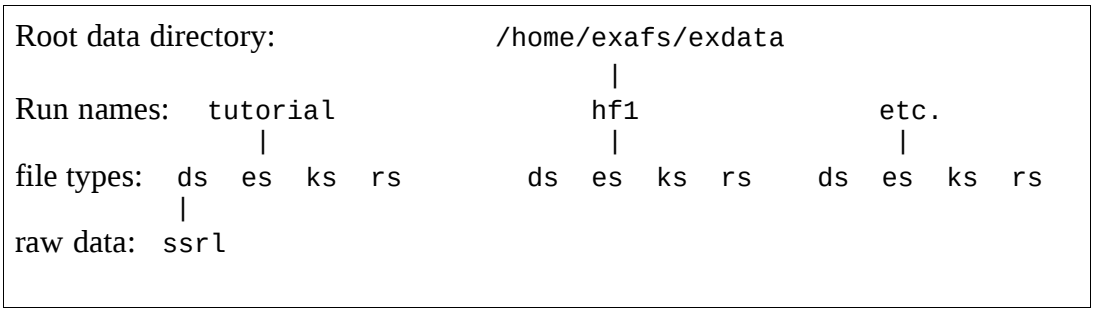
```
EXAFS_HOME=/home/exafs
PATH=./:$HOME/bin:$EXAFS_HOME/bin
EXDATA_DIR=$EXAFS_HOME/exdata
FITDIR=..

export PATH EXAFS_HOME EXDATA_DIR FITDIR
```

B. The File System

The programs can be run from any directory you wish to work in. However, the data files must be set up in a predefined directory tree. The root data directory is usually something like

'/home/exafs/exdata', and can be reached by the standard alias 'xxd'.

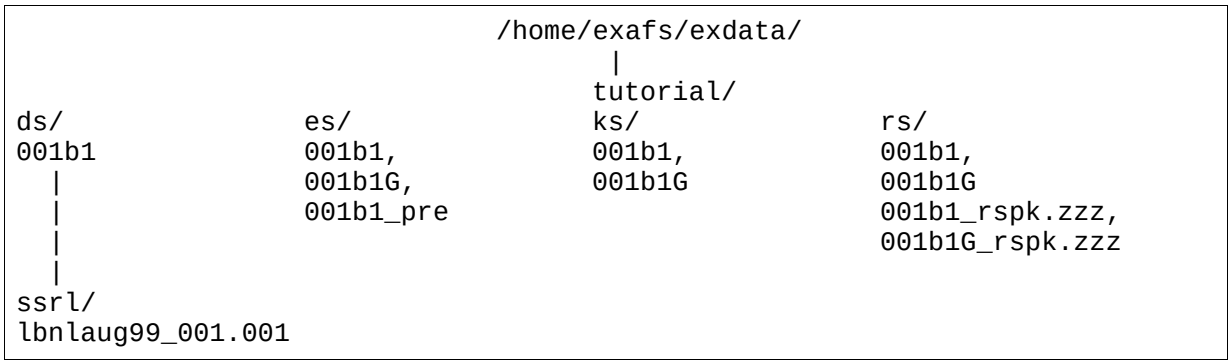
The tree will look something like this:



At each step of the analysis, you will "graduate" data from a lower level into an upper level, culminating in the r-space file. A typical procedure look like:

1. convert raw data in SSRL directory into d-space (data space) and possibly e-space format with CONVGERM. Files in ds/ and es/ are stored with their run numbers as their name, with the block number appended as in "001b1".
2. If data includes multi-channel fluorescence data, use SUG (SUGlitch) to make an e-space fluorescence file, with a name like "001b1G".
3. Use SUREDUCE to make a pre-edge-subtracted file, with a name like "001b1_pre" located in the /es subdirectory. This step should not necessarily be performed on fluorescence data, as will be explain in Sec. ???
4. Use SUREDUCE to determine a proper mu_0 background and produce k-space and r-space files with the "_pre" removed from the filename, ala "001b1" or "001b1G" under the subdirectories /ks and /rs respectively.
5. Use RSFIT to fit the data in r-space. The actual data file used is the k-space file, which RSFIT will transform. If desired, the fit function found will be stored in r-space as something like "001b1_rspk.zzz".

So in the end, you might have the following files in your directory tree:



A note about the file format: The data in the ssrl/ directory is in ascii, however, the rest of the data are stored in a binary format. To convert an /es, /ks, or /rs file to ascii, run BIN2GNU. To get back, run GNU2BIN. The format is not binary compatible between Intel and Sun processors, but is quickly converted using FIXENDIAN. Be VERY careful using FIXENDIAN with wildcards... please talk to me before trying this.

C. Preparing to reduce data (CONVGERM)

The first step is to convert the data from the raw, ascii data that comes from SSRL into the d-space and e-space files. This procedure is usually only done once at the end of a data run.

```
>convgerm

input subdir for exafs data files: <directory>
Save Lytle (Iff) fluorescence data [RET=N]? <RET>
Keep reference file data? [RET=y] <RET>

input-file PREFIX name?    i.e. Input sc6 for sc6_***.
<directory>
Input the start number
 <1>
Input the stop number
 <max sample number>
```

Here <**directory**> is simply the subdirectory in the exafs data directory that houses the SSRL ascii data. Input-file PREFIX name refers to the prefix to the SSRL ascii file. It is the same as the directory you saved your data to at the SSRL beamline, and is usually kept the same when the data is transfered back to the home server. We then select the starting sample number and some number which is greater than or equal to the number of sample runs you did during the experiment (any numbers larger than the maximum sample run number will be ignored).

D. PLOT

PLOT is simply a general purpose plotting program for looking at RSXAP-binary data. Several of the utility programs act like plot in one way or another, so let's start by running this program to look at some data. Type "plot" and you will see:

PLOT-Aug-1999

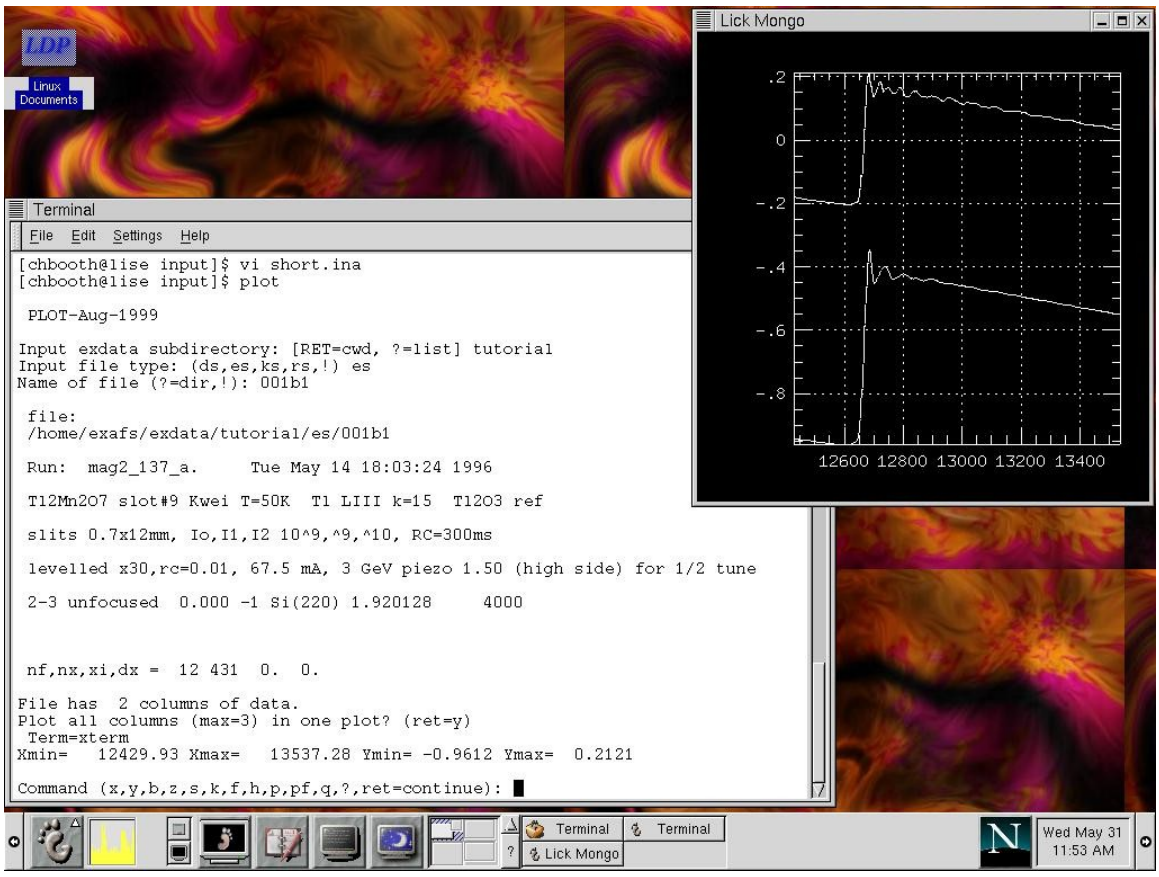
Input exdata subdirectory: [RET=cwd, ?=list]

PLOT is asking for your input about the file location. It will need the run (exdata) directory, the "type" of data (e-space, k-space, r-space) and the filename. Answer each question thusly:

```
PLOT-Aug-1999
```

```
Input exdata subdirectory: [RET=cwd, ?=list] tutorial
Input file type: (ds,es,ks,rs,!) es
Name of file (?=dir,!): 001b1

 file:
 /home/exafs/exdata/tutorial/es/001b1
 Run:  mag2_137_a.     Tue May 14 18:03:24 1996
 Tl2Mn2O7 slot#9 Kwei T=50K  Tl LIII k=15  Tl2O3 ref
 slits 0.7x12mm, Io,I1,I2 10^9,^9,^10, RC=300ms
 levelled x30,rc=0.01, 67.5 mA, 3 GeV piezo 1.50 (high side) for 1/2 tune
 2-3 unfocused  0.000 -1 Si(220) 1.920128     4000

 nf,nx,xi,dx =  12 431  0.  0.

File has  2 columns of data.
Plot all columns (max=3) in one plot? (ret=y) <RET>
```

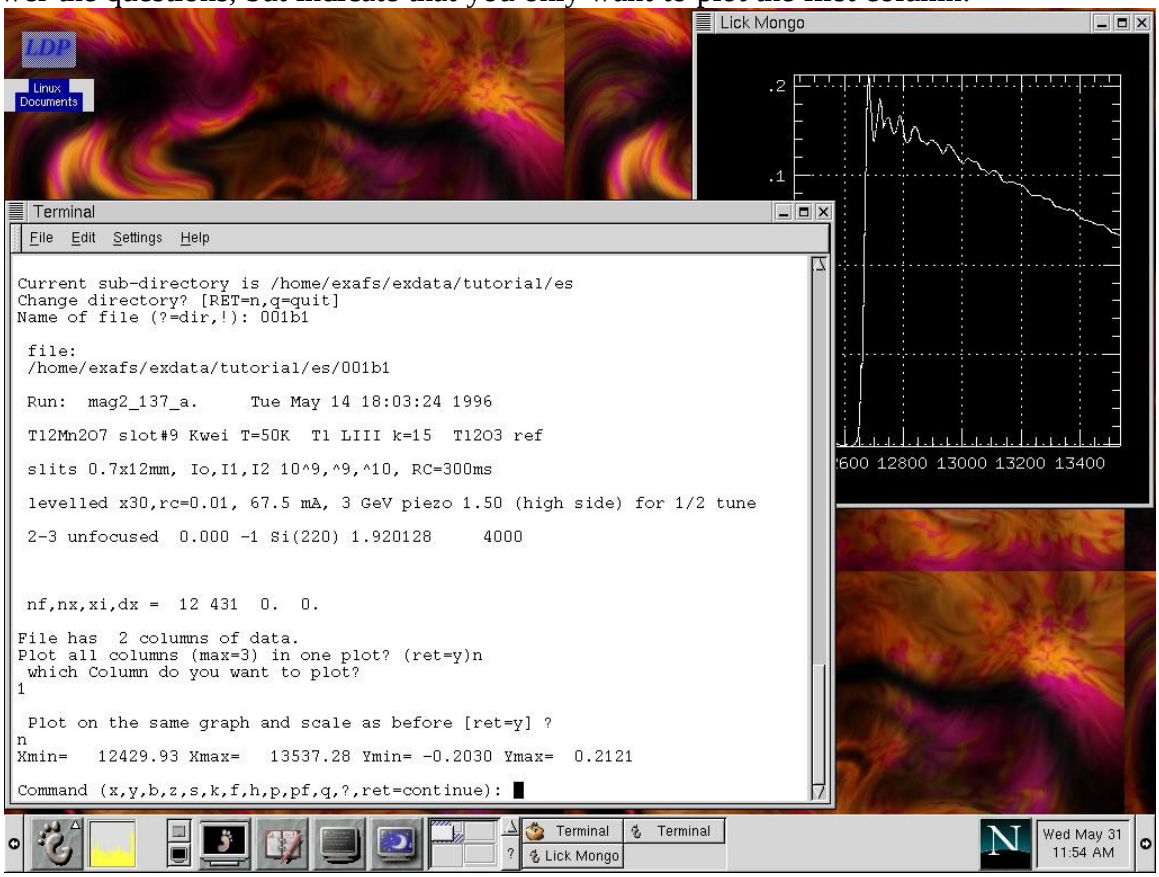
This last question is as it says... E-space files that have not had any other processing and have the reference material data (The one on the I2 detector) in a second column. For now, type <RET> to continue. Your screen will look like:



The "Command" line has many option. First of all, to continue to another file you could hit <RET>... DONT YET! Just type "?":

```
Command (x,y,b,z,s,k,f,h,p,pf,q,?,ret=continue): ? <RET>
x,y,b: Change scale
z: zoom with mouse
s:     Save the last file (not r-space)
c :     Toggle plotting curve or points
k :     Change k-weight of chi (k-space only)
f :     Make a fresh plot
h :     Plot cross-hairs
p :     Print a copy
pf:     Print a copy to a postscipt file 'temp.ps'
q :     quit to system

Command (x,y,b,z,s,k,f,h,p,pf,q,?,ret=continue):
```

Feel free to try a few of these. Notice that the plot has two lines on it. It isn't clear which is the reference material and which is the data. To plot just the data, hit return to continue, answer the questions, but indicate that you only want to plot the first column:



Notice the question:

```
Plot on the same graph and scale as before [ret=y] ? n <RET>
```

At this point I answered "n", which resets the plot and scale. To quite, type "q".

E. REDUCE

Now we are ready for some nitty gritty! In this example, I assume we are using data collected in the standard "transmission" geometry, that we have Tl LIII edge data in a file called 001b1 under the run directory "tutorial". We will copy this file to a filename with YOUR name so that you won't overwrite someone else's work, so before we get going, type the following:

```
$pushd /home/exafs/exdata/tutorial/es
$cp 001b1 corwin
$popd
```

where "corwin" is your name or some other unique filename. The "push" and "pop" commands just take you to the appropriate directory and then back to where you started.

Also, you should set up the directory structure for performing your work. Most of such organization is up to you, but the RSFIT program wants to run in a directory input or output and requires that such directories exist. Also, the REDUCE program uses a file comdata to store what your last commands have been, so it is a good idea to always run REDUCE from the same directory for a given project. The comdata file can be copied for back up purposes. To set up a directory tree like the one I use, type the following:

```
cd <RET>      This will take you to your home directory
mkdir Fit
cd Fit
mkdir tutorial
cd tutorial
mkdir reduce
mkdir input
mkdir output
mkdir structures
cd reduce
```

The data we will use in our example is Tl LIII edge EXAFS data from Tl2Mn2O7. The near neighbor structure includes 2 Tl-O neighbors at ~2.15 Ang., 6 Tl-O's at ~2.45 Ang, 6 Tl-Tl and 6 Tl-Mn at ~3.50 Ang., etc. The data were collected on BL 2-3 at SSRL back in 1996.

The reduction process involves the following steps:

1. Removing the pre-edge
2. Determining E0
3. Setting up a rough mu_0 background with either splines or polynomial
4. Picking transform properties
5. Fitting to the low-r part of the transform to determine final mu_0
6. Saving all the results!

1. Removing the pre-edge

Start by running REDUCE. You will see a screen like:

```
**************************************************
************Welcome to REDUCE 1.12x****************
**************************************************
*    If you have any questions or comments, please  *
*    send email to Corwin Booth (chbooth@lbl.gov)   *
******************Good luck************************
**************************************************

**************************************************
*                                                *
*                Input file properties           *
*                                                *
**************************************************

1. Next process                   : bg and fft
2. exdata subdirectory            :
3. Input file                     :
4. Collection mode                : transmission
11. Plot data file                : n
12. K-weight                      : 1
Command (#, (h)eader, (q)uit, [continue=RET]):
```

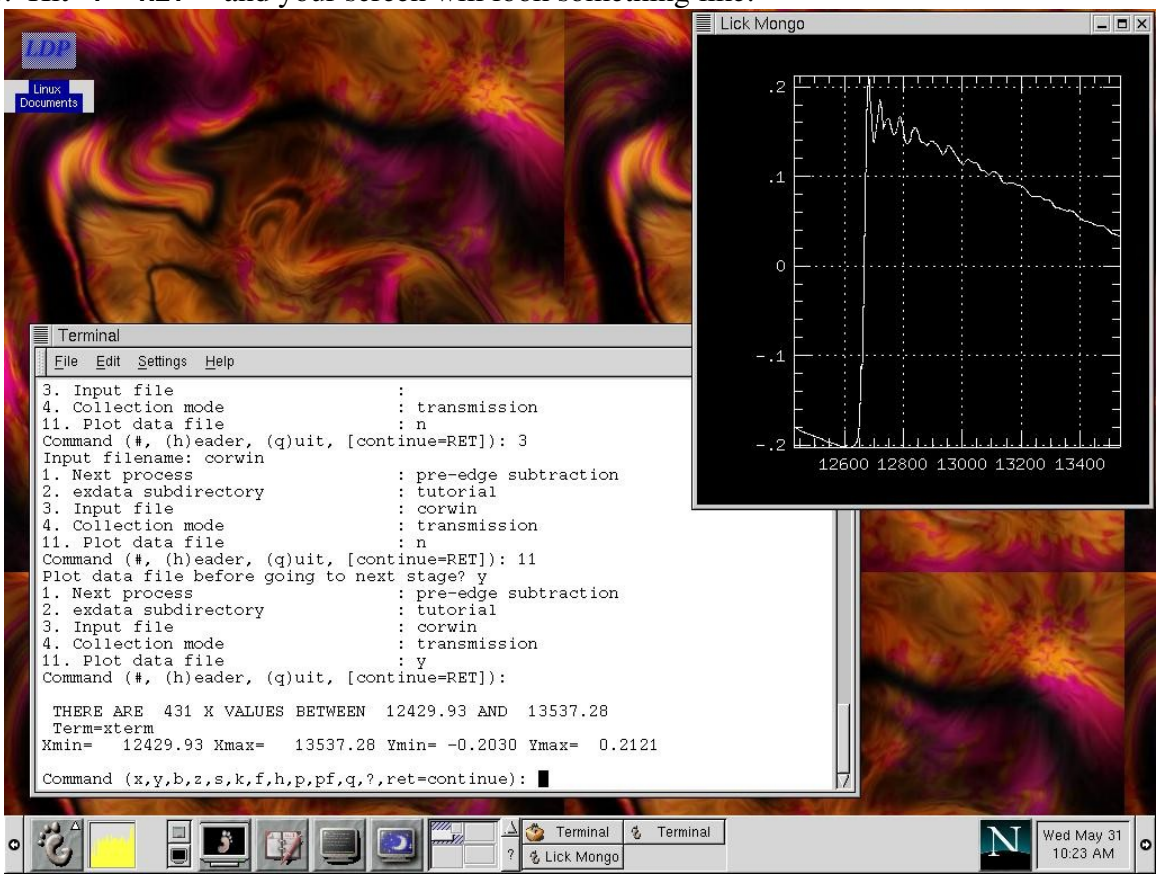
This is the first of the main menus (there are five). Here we tell REDUCE what we want to do with the data, where the data is stored, and some information about the data. Type '**1 <RET>**' and you will be told what to do:

```
transform a ks file (tk)
transform an rs file (tr)
remove post-edge background (b)
subtract pre-edge (p)
remove post-edge bg then transform (a)

Input process:
```

Type '**p <RET>**'. This menu updates automatically. Then type '**2 <Ret>**' and input the run name of '**tutorial**', then '**3**' and the run number. The collection mode is transmission, so we'll leave that alone. We'll plot the data, too. Once all this is entered, the menu will look like:

```
1. Next process                   : pre-edge subtraction
2. exdata subdirectory            : tutorial
3. Input file                     : corwin
4. Collection mode                : transmission
11. Plot data file                : y
Command (#, (h)eader, (q)uit, [continue=RET]):
```

If you want to see the run header before going on, type 'h', otherwise, hit RETURN. You will then see a plot of the data appear in a separate window, with a command line, from which you can scale the plot or hit RETURN to continue. Do so! A plot will appear with a "`Command:`" line. Hit "**`? <RET>`**" and your screen will look something like:



You should play with the different plot commands, such as the rescaling commands 'x' and 'y'. To continue, hit <RET> again. (Getting the idea?)

```
THERE ARE  431 X VALUES BETWEEN  12429.93 AND  13537.28
 *** In sbgfit***

*****************************************************
*                                                   *
*       Properties of PRE-edge subtraction          *
*                                                   *
*****************************************************

 1. Input edge [eg. Pr LIII]                 : Cu K
 2. Normalize by edge height [y,n]           : y
 3. Emin for pre-edge fit                    :  8800.0 eV
 4. Emax for Pre-edge fit                    :  8900.0 eV
 5. Emin for Eo fit                          :  9100.0 eV
```

```
 6. Emax for Eo fit                          :  9400.0 eV
 7. Energy above Eo to start Vic fit         :   500.0 eV
 8. End energy for Vic (0=top)               :     0.0 eV
 9. Power of E for Cby                       : -1.0
10. Order of Cby                             : 4
11. Plot original data [y,n]                 : n
12. Plot fit [y,n]                           : n
13. Plot subtracted data [y,n]               : n
14. Save subtracted data [y,n]               : n
Command (#, (f)resh menu, (q)uit, [continue=RET]):
```

First you should notice that before the menu banner was printed, REDUCE told you a little bit about the input file, namely, the number of data points and the energy range. This menu reports the parameters for removing the pre-edge absorption, and the defaults are for a Cu *K* edge data set. Therefore, we have to change everything for our Tl L_{III} edge data.

Before we continue, we need to briefly discuss the methodology for removing the pre-edge used by REDUCE. The basic idea is to end up with a pre-edge-subtracted data set that has the correct average energy dependence for the measured edge. The raw data is not this way because there are many other absorption processes going on besides the one associated with the absorption edge we are measuring. These processes include the background absorption from the other elements in the material, the lower energy processes from the species we are measuring (for instance, the M edges), the gas in the ion chambers, the material used for the windows on the sample holder, etc. etc. Because we normally don't want to keep track of all these things, REDUCE uses a Victoreen formula to force the decay of the absorption above the edge to follow the formula:

$$\mu = A\,\lambda^3 + B\,\lambda^4 \quad ,$$

where *A* and *B* are unique for each edge of each absorbing atomic species. *A* and *B* are tabulated in Teo's EXAFS book, and these values are used by REDUCE.

In order for this formula to work, REDUCE needs to know the threshold energy *E*0 and the step height (aka the change in the absorption at the threshold energy). REDUCE uses the "half-height" approach for finding *E*0 by fitting a line through a portion of the data above the edge and a low order polynomial below the edge to obtain the step height, and then finds the energy where the step height is ½ of maximum. REDUCE therefore needs you to give it the range over which to fit the pre-edge part of the spectrum to determine the base, and the range over which to fit the post-edge part to extrapolate the edge step height and E0. From Fig. 1, it looks as if the absorption edge starts just above 12650 eV, and that a linear fit from 12700 to 13000 eV will cross the edge at about the right height. For the lower energy for the pre-edge, we'll just put in '0' so that REDUCE will use the lowest energy available. Plug in the numbers, and hit "f <RET>" to check your work:

```
 ***************************************************
*                                                   *
*       Properties of PRE-edge subtraction          *
*                                                   *
***************************************************
```

```
 1. Input edge [eg. U LIII, no edge=free fit]: Tl LIII
 2. Normalize by edge height [y,n]           : y
 3. Emin for pre-edge fit                    :     0.0 eV
 4. Emax for Pre-edge fit                    : 12600.0 eV
 5. Emin for Eo fit                          : 12700.0 eV
 6. Emax for Eo fit                          : 13000.0 eV
 7. Energy above Eo to start forced fit      :   200.0 eV
 8. End energy for forced fit (0=top)        :     0.0 eV
 9. Power of E for Cby                       : -1.0
10. Order of Cby                             : 4
11. Plot original data [y,n]                 : y
12. Plot fit [y,n]                           : y
13. Plot subtracted data [y,n]               : y
14. Save subtracted data [y,n]               : n
Command (#, (f)resh menu, (q)uit, [continue=RET]):
```

Then hit <RET> to continue. At this point, the subtraction has been performed, and we are plotting the results to check our work. REDUCE will spit out the following:

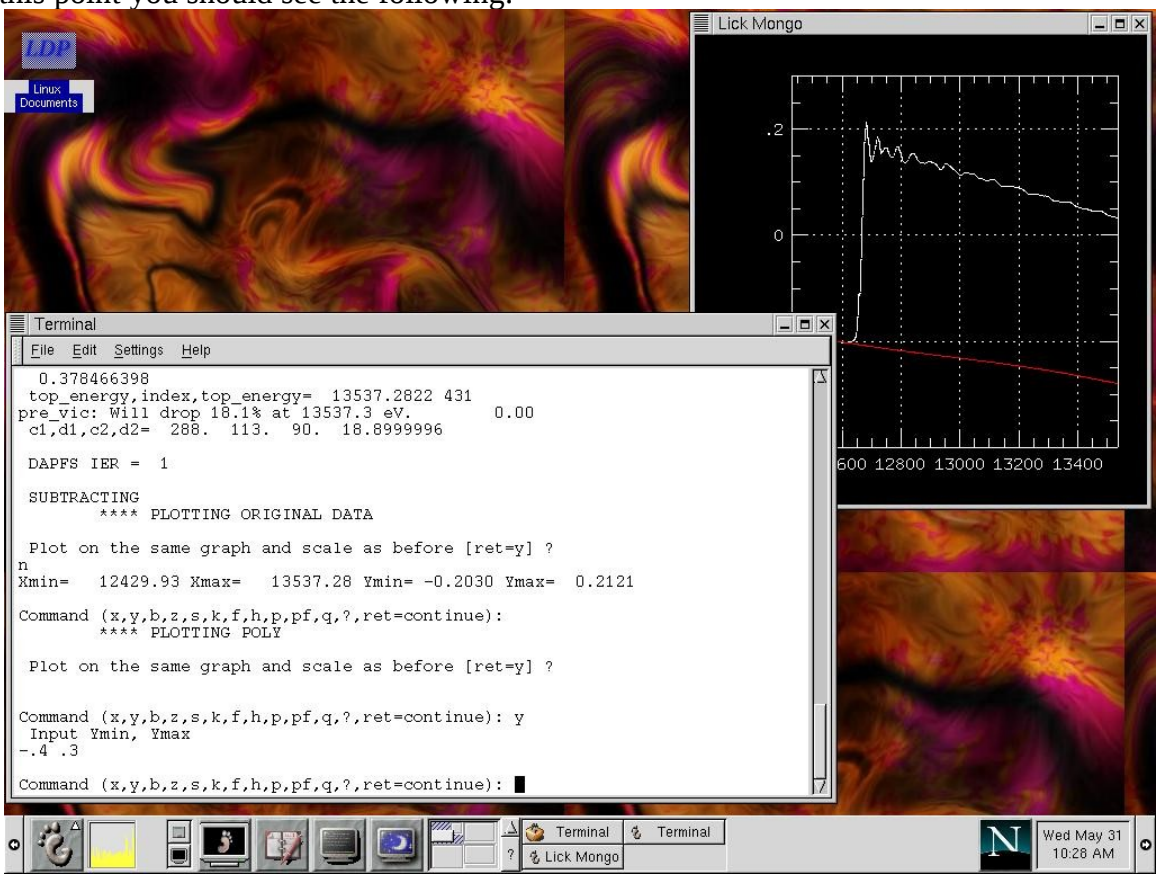
```
WEIGHTED FIT TO POLY IN Y = (E-E0)**P
 with INT POWER OF SQRT(E-E0) IN WEIGHT
  [TYPICAL ABOVE THRESHOLD CHEBYSHEV: P=0.5 AND WEIGHT=2 ]

    17 PTS INCLUDED,  NMAX2= 2004
 in pre_vic...
 pre_vic: reset x_min for baseline fit to   12429.9346
 Xmin=  12429.9346  12429.9346  Xmax=  12589.9268  12600.
 slope,intercept= -0.000129667824  1.43054879
 Xmin=  12699.6113  12700.  Xmax=  12998.0254  13000.
 slope,intercept above edge= -0.000141215642  1.95767653
Eo,Yo=12665.890.16768
 e_not,height,base,edge_height=  12665.8916 95  0.167679876 -0.210786521
  0.378466398
 top_energy,index,top_energy=  13537.2822 431
pre_vic: Will drop 18.1% at 13537.3 eV.        0.00
 c1,d1,c2,d2=  288.  113.  90.  18.8999996

 DAPFS IER =  1

 SUBTRACTING
        **** PLOTTING ORIGINAL DATA

 Plot on the same graph and scale as before [ret=y] ?
```

Most of this text is information about how the pre-edge removal went, which you are free to ignore. This information might include a warning that REDUCE used the Victoreen coefficient for Bi if Z for the element you are studying is > 83, because the tabulation in the codes only goes that high (this will be corrected in the near future, but it works pretty well anyway). The last line is a question that asks whether you want to plot on the same scale as before. That is because there is already a plot on the screen. We are just replotting the same data, but give it a new scale anyway, so type "**n** <**RET**>". Then continue at the "Command" line and answer the next question about the scale by just hitting "**<RET>**". This will plot the pre-edge on the same plot as the data.

So why can't you see the pre-edge fit? It is off scale below the plot window, so rescale the plot by typing at the "`Command:`" prompt:

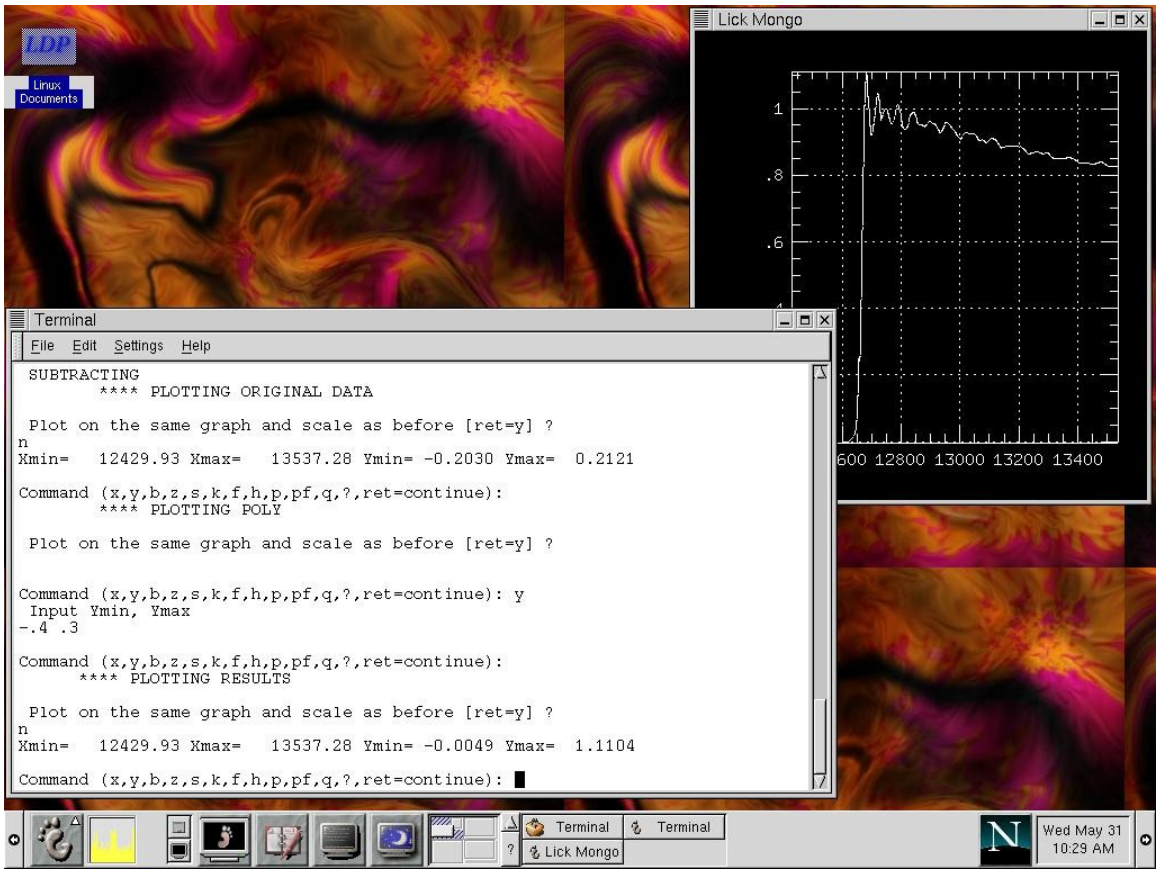
`y <RET>`
`-.4 .3 <RET>`

At this point you should see the following:



At this point it is good to evaluate the quality of the fit, especially in the pre-edge region. Does the fit go through the data nicely below the edge? When the fit gets into the edge region, does it rise too quickly and actually cross the data, or remain tangent to the pre-edge region? This fit looks very good at this point.

Now continue and plot the final, subtracted data. Put this data on a new scale because it will be normalize to unity at the edge step. Your screen will look something like this:



At this point, once again we need to evaluate by eye the final subtracted result. The main thing here is to look at where REDUCE thinks the edge step is. Since we normalized the data, that is where the data first crosses unity. This fit looks pretty good, but imagine if this point was 5% higher. That would be unacceptably bad, and so you might have to go back and play with the "Emin for E0" and "Emax for E0" quantities on the pre-edge menu.

Hit **<RET>** to continue back to the first menu. The whole process starts again here. Since we like this fit, you should do the following:

1. Turn the plotting off at the "Input file properties" menu (Item 11).
2. If you want, you can turn the plotting off at the "PRE-edge subtraction" menu. If you have other Tl edge data to do, I'd just turn off Items 11 and 12.
3. Turn ON the "Save subtracted data" (Item 14).
4. Return to the first menu after saving the data into the file "corwin_pre".

Congratulations! You have completed Step 1 and done a rough job on Step 2 in the Procedures. Once all these items for pre-edge removal of your Tl data have been input, it is a good idea to remove the pre-edge from all similar data that you have. This part goes very fast... mostly just hitting return to continue and changing the data file name.

2. Obtaining E_0

This is already done, more or less, but we will do it again in part 3.

3. Doing a post-edge background removal

The process of determining the mu_0 background is relatively quick, however keep in mind that the first steps involve getting reasonable starting parameters for the fit, the next steps involve optimizing one of those parameters "automatically", and the final steps are merely checking the sensitivity of the final result to the starting parameters. This process will involve looking at the data in e-space, k-space and r-space, and looking at the chosen background function. From the "Input file properties" menu, make the menu look like this:

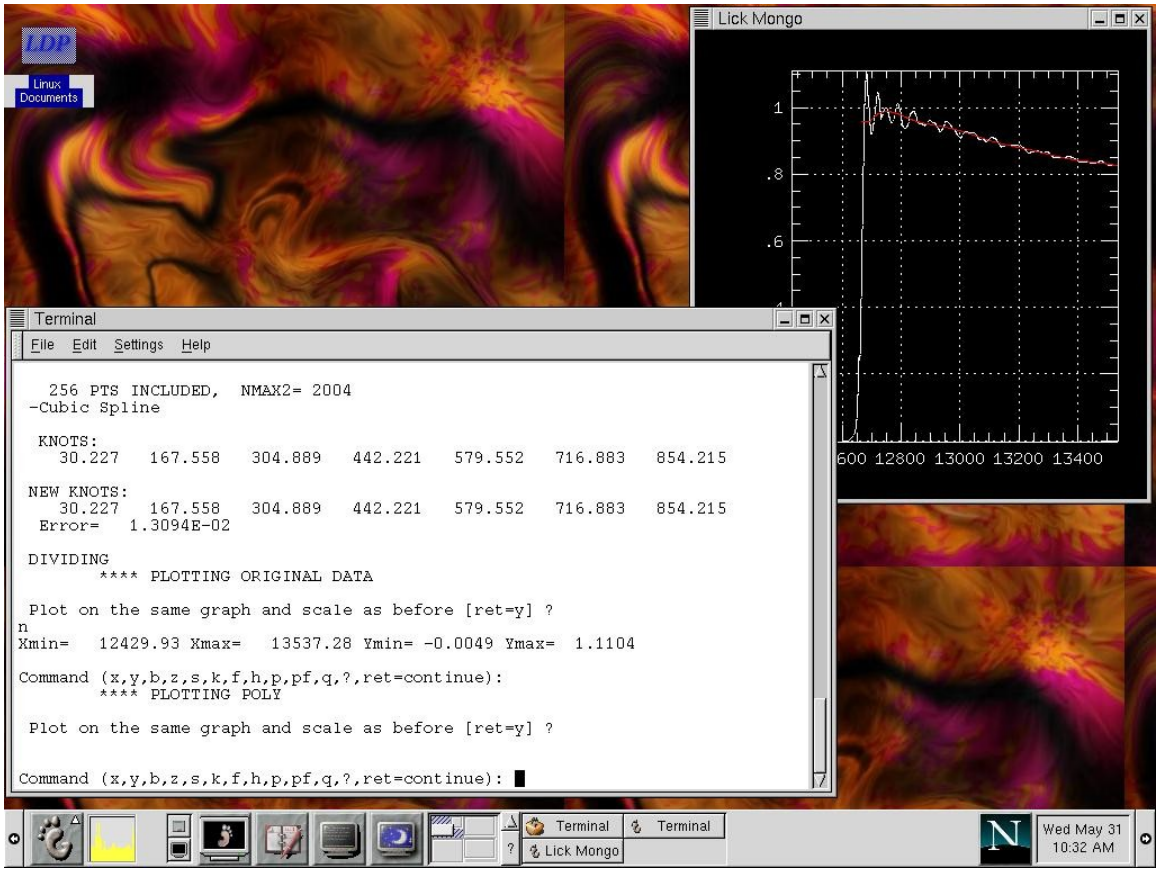
```
*****************************************************
*                                                   *
*                Input file properties              *
*                                                   *
*****************************************************

1. Next process                    : bg and fft
2. exdata subdirectory             : tutorial
3. Input file                      : corwin_pre
4. Collection mode                 : transmission
11. Plot data file                 : n
12. K-weight                       : 1
Command (#, (h)eader, (q)uit, [continue=RET]):
```

Item 1 was made by typing "`1 <RET> a <RET>`". We will use a k weight of one, so just hit `<RET>`:

```
THERE ARE  431 X VALUES BETWEEN  12429.93 AND  13537.28
 *** In sbgfit***

*****************************************************
*                                                   *
*       Properties of POST-edge subtraction         *
*                                                   *
*****************************************************

 1. Determine Eo by linear fit (Toggle) ?    : y
 2. E0 min for linear fit                    : 12700.00
 3. E0 max for linear fit                    : 13000.00
 4. Use Kmax(K) or Emax for upper fit range  : k
 5. Kmax for post-edge fit                   : 15.00
 7. Power of (E-Eo) for post-edge fit        :  1.00
 9. # of knots for spline [for spline only]  :    7
10. spline or (c)heby (Toggle) ?             : S
11. Eo, the Energy (eV) for the edge         : 12665.89
12. Emin for post-edge fit                   : 12695.89
```

```
14. Plot original data?                          : n
15. Plot fit?                                    : n
16. Plot results ?                               : n
17. Save results?                                : n
 Which one would you want to change?
 ([RET] next menu, (f)resh display, (a)dvanced options)
```

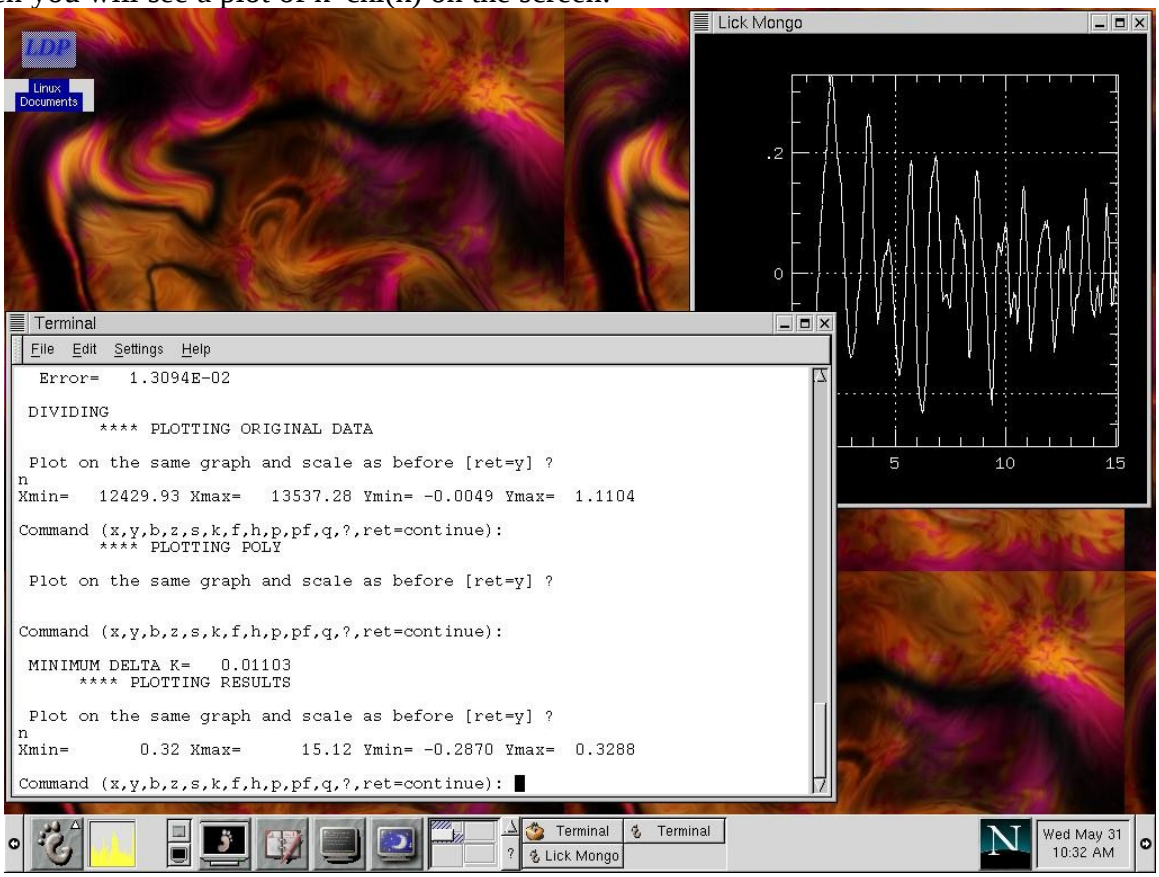
At this point we will make some choices about the background. First, let's start with the defaults. These include using a spline function for the background (item 10) with 7 knots (item 9) placed evenly in energy (item 7) up to a k (item 4) of 15.0 inv Ang (item 5, and given by E and E0, which is in item 11).Change all the plotting on (make 14,15,16 all "y"), and continue by hitting <RET>. Go ahead and plot the original data, then continue to plot mu_0. At this point the screen should look like:



Among other things, notice that the knot positions are typed on the screen. Now that we have E0 and u0, we can calculate chi(k)=(mu/mu_0) - 1 and k=0.512 (E-E0)^1/2. Hit <RET> and give the next plot (the "result") a new set of axes by answering "n" to the question:

```
      **** PLOTTING RESULTS

 Plot on the same graph and scale as before [ret=y] ?
n
```

Then you will see a plot of k*chi(k) on the screen:



Why k*chi(k) and nor chi(k) or k^3*chi(k)? Well, this is something of a religious question in EXAFS circles. These codes come from a group of people that felt that "k-weighted" data made the most sense, because the EXAFS oscillations need a "boost" at high k (the data decay as 1/k in the absense of any disorder) BUT not too big a boost, because usually the data quality at high k is not as good as at low k. However, if you have a system with a moderate amount of disorder (almost any real system), the oscillations decay closer to 1/k^3, so another school said, "well, the data at high k isn't as high quality, but in the presence of disorder, we need that data anyway." The main take-home point at the moment on this subject is that these codes and the fit codes that follow like to start with k-weighted data. You can fit the data however you want, but the data are actually saved as two column data with k vs. k*chi.

Anyway, now we will continue to do the Fourier transform (FT). Hit <RET> until you see the following:

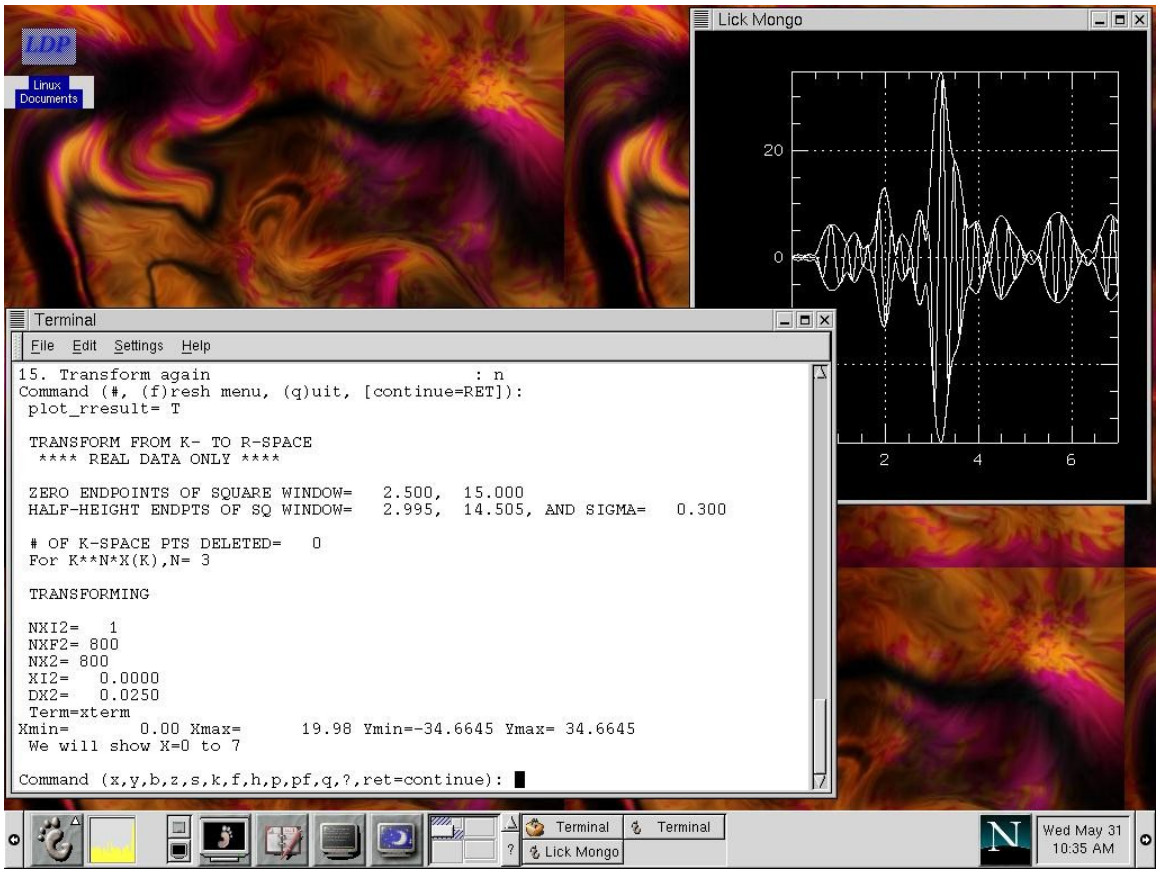
```
*****************************************************
*                                                   *
*        Properties of Fourier transform            *
*                                                   *
*****************************************************
```

```
 1. Change K-weight by                       : 0
 2. K-window minimum                         :  3.0
 3. K-window maximum                         : 16.0
 4. K-window sigma                           :  0.3
 5. K-space shift (in eV)                    :  0.0
 6. For R to K, divide by K-window           : y
 7. R-window minimum                         :  0.0
 8. R-window maximum                         :  7.0
 9. R-window sigma                           :  0.1
10. R-space shift (in Ang)                   :  0.0
11. Plot shift results                       : n
12. Plot transform results                   : y
13. Save transform results                   : n
14. Retain B1,B2                             : n
15. Transform again                          : n
Command (#, (f)resh menu, (q)uit, [continue=RET]):
```

Here we select the properties of the FT window and the k-weighting *of the transform only*. Looking at the plot of the data on the previous screen shot, you should notice that the data only extend out to just beyond a k of 15. We will use the FT of k^3*chi, so begin by changing item 1 to "2", that is we will increase the power of k in the k-weight by 2, from 1 to 3. Also change the Kmax of the window to 15 and the Kmin of the window to 2.5. The menu should look like this:

```
 1. Change K-weight by                       : 2
 2. K-window minimum                         :  2.5
 3. K-window maximum                         : 15.0
 4. K-window sigma                           :  0.3
 5. K-space shift (in eV)                    :  0.0
 6. For R to K, divide by K-window           : y
 7. R-window minimum                         :  0.0
 8. R-window maximum                         :  7.0
 9. R-window sigma                           :  0.1
10. R-space shift (in Ang)                   :  0.0
11. Plot shift results                       : n
12. Plot transform results                   : y
13. Save transform results                   : n
14. Retain B1,B2                             : n
15. Transform again                          : n
Command (#, (f)resh menu, (q)uit, [continue=RET]):
```

Hit <RET> to continue, and you will see your first FT!



The way the code plots transforms might be new to you. The outer envelope is the magnitude of the FT, that is the (Re^2+Im2)^1/2 of the complex transform. This envelope is also plotted as the negative of the amplitude. The oscillating line in between the envelope is the real part of the transform. Details of this real part are useful for identifying the species of backscattering atom and other properties, but for now, just focus on the amplitude.

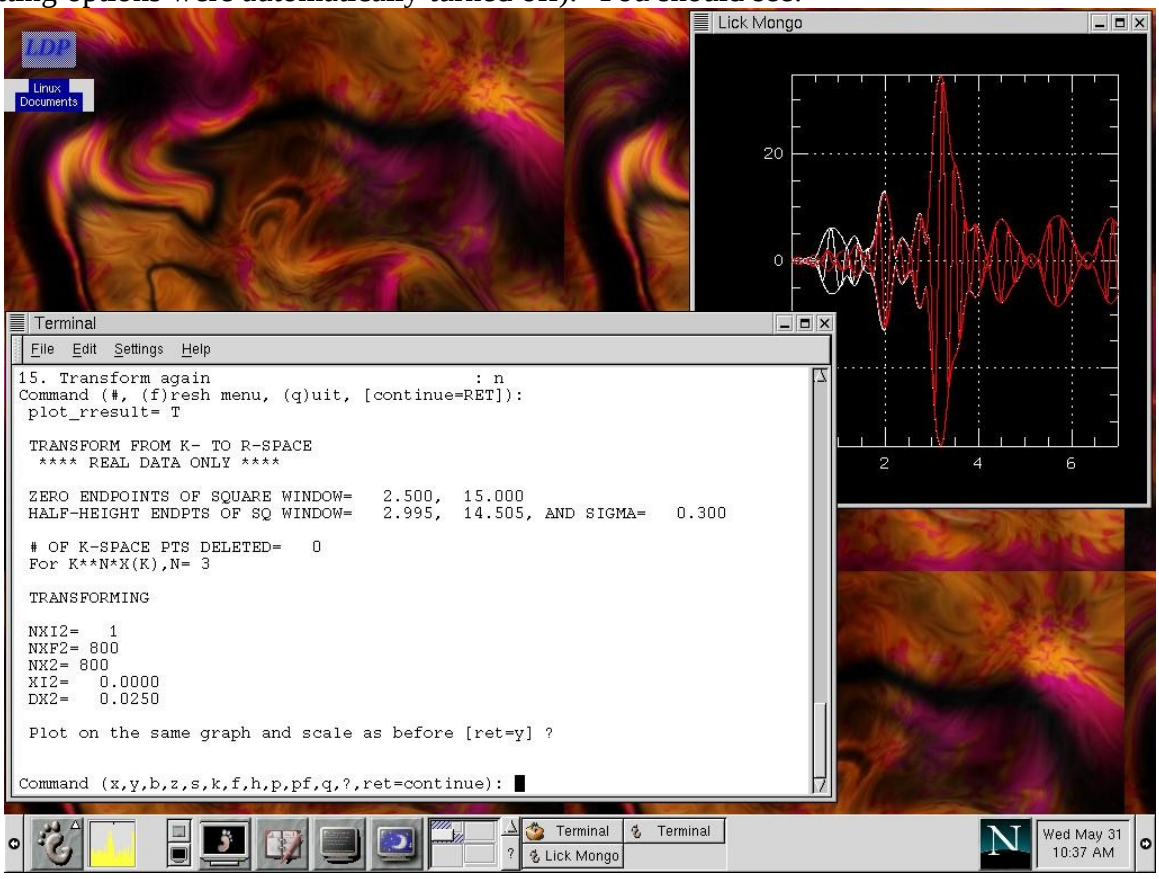
Now notice that there is a minor peak at about 1 Ang, followed by the main peak at about 2.0 Ang. The minor peak at 1 Ang is NOT due to Tl-O backscattering! It is due to a poor (but actually, not terrible) background. We will now optimize this background. Hit <RET> and you will see:

```
*****************************************************
*                                                   *
*     Properties of automatic Emin determination    *
*                                                   *
*****************************************************

 Choose Emin fitting option:
 y0: minimize the amplitude area in low r region
 y1: fit low r amplitude to straight line
 y2: fit low r amplitude to quadratic line
 q : quit REDUCE
```

Follow this procedure:

```
y0
Input Rmax for search range [RET= 1.2]:<RET>
Input Energy search range [RET=12676,12696]:<RET>
 Final Evar, aaa,bbb,ccc =  12682.5723  0.  0.  0.

*****************************************************
*                                                   *
*     Properties of automatic Emin determination    *
*                                                   *
*****************************************************

 Choose Emin fitting option:
 y0: minimize the amplitude area in low r region
 y1: fit low r amplitude to straight line
 y2: fit low r amplitude to quadratic line
 q : quit REDUCE
```

Now cycle through the menus by hitting <RET> and replot the new transform (all the other plotting options were automatically turned off). You should see:



Notice the "bad" peak is reduced. This data may need more tuning however. Try changing the number of splines, etc. The "rules" of a good background are to

1. Most of the changes in the FT when choosing a new background will be in the first peak. For materials with short bond lengths, getting the right background is critical to getting the right amplitude! You must find a background that doesn't change the amplitude of the first peak from whatever the "right" answer is. This is, sadly, a black art.
2. Make sure the background is not overly sensitive to the parameter choices used to determine it. For instance, going from 7 to 6 or 8 splines should not have a big difference on the amplitude of the first peak.
3. While not affecting the first peak amplitude, you want to reduce lower-r peaks. In this data, the procedure is somewhat simplified by the long nearest neighbor Tl-O peak because you can simply try to minimize the amplitude in the low-r region (choice "y0" in the auto-Emin determination). Sometimes "y2" is more appropriate, and for particularly tricky cases, one may employ more advance methods (the "iterative background determination" described in F. Bridges, C. H. Booth, and G. G. Li, Physica B 208&209, 121 (1995), and hopefully in future versions of this manual).

F. RSFIT

RSFIT is an r-space fitting program. The program fits the sum of "standard" files of (typically) individual scattering paths to the data set. These standard files can be generated either from a theoretical calculation such as is performed by the FEFF code from University of Washington, or can be made from a real data set on a well understood compound.

1. Making a reference standard with ATOMS and FEFF

The first step in making a fit is to decide what sort of standard you wish to use. We will make standards with the FEFF code, using the crystal structure of Tl2Mn2O7 from Reitveld analysis of neutron diffraction data. You should probably take a look at the FEFF documentation, but we will step through the process here. First we need to make an input file for FEFF with an appropriate cluster of atoms. For this purpose we can use the program ATOMS (shipped with FEFF). Assuming you will use atoms, get into your "structures" directory by typing something like

```
cd ../structures
```

And edit a file "atoms.inp" to look like:

```
title Tl2Mn2O7 (Shimakawa et al, PRB 55, 6399 1997)
space O_H^7
a=9.89093
index=true rmax=7 core=tl
```

```
atom
tl  0.5    0.5    0.5     Tl
mn  0.0    0.0    0.0     Mn
o   0.3261 0.125  0.125   O(1)
o   0.375  0.375  0.375   O(2)
```

ATOMS expects this info to be in a file named "atoms.inp". Once this file is created, type "atoms" and a file "feff.inp" will be created. ATOMS will give the following warning:

```
 ===============================================================================
 ATOMS 2.46b                                                    by Bruce Ravel
 ===============================================================================
   title > Tl2Mn2O7
 *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 Space group f d 3 m is commonly referenced with alternative origins.
 If you are displeased with the resulting atom list, shift all atomic
 coordinates in atom.inp by (-.125,-.125,-.125) and run atoms again.
 *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 Output written to feff.inp
 ===============================================================================
```

which in our case is OK. The "feff.inp" file starts out looking like:

```
 * This feff.inp file generated by ATOMS, version 2.46b
 * ATOMS written by Bruce Ravel and copyright of The Univ. of Washington, 1994
 *  Space group f d 3 m is commonly referenced with alternative origins.       *  If
you are displeased with the resulting atom list, shift all atomic       *
coordinates in atom.inp by (-.125,-.125,-.125) and run atoms again.
 * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- *
 *       total mu =     1059.5 cm^-1, delta mu =      541.2 cm^-1
 *       specific gravity =  8.658, cluster contains  129 atoms.
 * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- *
 *       mcmaster corrections:  0.00038 ang^2 and  0.658E-06 ang^4
 * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- * -- *

 TITLE   Tl2Mn2O7 (Shimakawa et al, PRB 55, 6399 1997)

 HOLE 4   1.0     Tl L3 edge ( 12.657 keV), s0^2=1.0

 *         mphase,mpath,mfeff,mchi
 CONTROL   1      1     1     1
 PRINT     1      0     0     0

 RMAX     6.99395

 *CRITERIA     curved   plane
 *DEBYE        temp     debye-temp
 *NLEG         8

 POTENTIALS
 *   ipot   z  label
       0   81   Tl
       1    8   O
       2   25   Mn
       3   81   Tl
```

```
ATOMS
   0.00000     0.00000     0.00000    0   Tl_00            0.00000
   1.23637     1.23637     1.23637    1   O(2)_01          2.14145
  -1.23637    -1.23637    -1.23637    1   O(2)_01          2.14145
   1.72003    -1.23637    -1.23637    1   O(1)_01          2.45270
  -1.23637    -1.23637     1.72003    1   O(1)_01          2.45270
   1.23637    -1.72003     1.23637    1   O(1)_01          2.45270
  -1.72003     1.23637     1.23637    1   O(1)_01          2.45270
   1.23637     1.23637    -1.72003    1   O(1)_01          2.45270
  -1.23637     1.72003    -1.23637    1   O(1)_01          2.45270
  -2.47273     0.00000     2.47273    2   Mn_01            3.49697
  -2.47273    -2.47273     0.00000    3   Tl_01            3.49697
   0.00000    -2.47273    -2.47273    3   Tl_01            3.49697
   2.47273     0.00000     2.47273    3   Tl_01            3.49697
   0.00000     2.47273    -2.47273    2   Mn_02            3.49697
  -2.47273     0.00000    -2.47273    3   Tl_02            3.49697
   0.00000    -2.47273     2.47273    2   Mn_03            3.49697
  -2.47273     2.47273     0.00000    2   Mn_03            3.49697
   0.00000     2.47273     2.47273    3   Tl_03            3.49697
   2.47273    -2.47273     0.00000    2   Mn_04            3.49697
   2.47273     2.47273     0.00000    3   Tl_04            3.49697
   2.47273     0.00000    -2.47273    2   Mn_05            3.49697
   0.75270    -1.23637     3.70910    1   O(1)_02          3.98153
  -1.23637     3.70910     0.75270    1   O(1)_02          3.98153
   1.23637    -3.70910    -0.75270    1   O(1)_02          3.98153
   3.70910     0.75270    -1.23637    1   O(1)_02          3.98153
```

As you can see, FEFF needs a cluster of atomic position to calculate the backscattering amplitudes. There is no reason that you can't type in this information by hand, but it is much simpler to use atoms to calculate the clusters for you. Lines that begin with a "*" are ignored and are used for comments. All of the commands are described in the FEFF documentation (type man feff7 for more info). The main one we will focus on here is

```
CONTROL  1  1  1  1
```

This command basically says "do a full calculation", as signified by the "1"'s. When you are starting a new calculation, you need to do a full calc. By the way, this is the default, so you could comment this line out.

Now type "feff7" and let feff do its thing. When it is done, it will output some important files:

chi.dat	This file contains the calculation of the EXAFS chi.
paths.dat	Lists each path and an associated path number
list.dat	Lists the path numbers used in the calculation

Now type "cp list.dat list.all" to make a backup. We will now edit list.dat to isolate a particular path to make a standard. list.dat looks like:

```
Tl2Mn2O7 (Shimakawa et al, PRB 55, 6399 1997)                    Feff  7.02
 Abs   Z=81 Rmt= 1.349 Rnm= 1.601 L3 shell
 Pot 1 Z= 8 Rmt= 0.863 Rnm= 1.069
```

```
 Pot 2 Z=25 Rmt= 1.052 Rnm= 1.322
 Pot 3 Z=81 Rmt= 1.335 Rnm= 1.578
 Gam_ch=5.235E+00 H-L exch
 Mu=-1.567E+00 kf=2.159E+00 Vint=-1.933E+01 Rs_int= 1.679
 ---------------------------------------------------------------------
  pathindex     sig2   amp ratio    deg    nlegs  r effective
        1     0.00000   100.000     2.000     2   2.1415
        2     0.00000   100.000     6.000     2   2.4527
        3     0.00000    42.280     6.000     2   3.4970
        4     0.00000    49.012     6.000     2   3.4970
        5     0.00000     5.725    12.000     3   3.7355
        6     0.00000     8.838    12.000     3   3.7753
        7     0.00000     5.470    12.000     3   3.8899
        8     0.00000    10.301    24.000     3   3.9266
        9     0.00000    49.655    12.000     2   3.9815...
```

Edit the file so that only the first path remains:

```
Tl2Mn2O7 (Shimakawa et al, PRB 55, 6399 1997)                    Feff  7.02
 Abs   Z=81 Rmt= 1.349 Rnm= 1.601 L3 shell
 Pot 1 Z= 8 Rmt= 0.863 Rnm= 1.069
 Pot 2 Z=25 Rmt= 1.052 Rnm= 1.322
 Pot 3 Z=81 Rmt= 1.335 Rnm= 1.578
 Gam_ch=5.235E+00 H-L exch
 Mu=-1.567E+00 kf=2.159E+00 Vint=-1.933E+01 Rs_int= 1.679
 ---------------------------------------------------------------------
  pathindex     sig2   amp ratio    deg    nlegs  r effective
        1     0.00000   100.000     2.000     2   2.1415
```

Now edit feff.inp so that the CONTROL line reads:

```
CONTROL  0  0  0  1
```

Now rerun feff (type "feff7").

OK, we are nearly done. The last thing we have to do is convert the chi.dat to the RSXAP binary format and place it into an appropriate file within the directory tree. Type "chi2bin", which will output:

```
Input file: chi.dat
 ****WATCH the PHASE!!!!********
 Output file to:

Input exdata subdirectory: [RET=cwd, ?=list]
```

We'll ignore that "`WATCH THE PHASE!!!!`" stuff for now. You need to give the exdata subdir. Here we'll use "hf1". We also want to store the file in a k-space directory and call the file something sensible, like "Mo_O2.02.f7". We will also elect not to add any lines to the header and normalize the amplitude to the number of neighbors in the path (the "deg", or degeneracy in the list.dat output). The I/O will look like this:

```
Input file: chi.dat
 ****WATCH the PHASE!!!!********
 Output file to:
```

```
Input exdata subdirectory: [RET=cwd, ?=list] tutorial
Input file type: (ds,es,ks,rs,!) ks
Name of file (?=dir,!): tl_o2.14.f7

file: /home/exafs/exdata//tutorial/ks/tl_o2.14.f7
 Read in chi.dat generated by Feff  7.02
 # of headline you want to add in
0

Normalize amplitude by the # of neighbours:2
   2.1415
 r1=  2.1415
  plot k*amp*r**2/n_pair
 Term=xterm
Xmin=       0.00 Xmax=      20.00 Ymin=  0.0000 Ymax=  0.3213

Command (x,y,b,z,s,k,f,h,p,pf,q,?,ret=continue):
```

At this point you could continue to look at the plots, or just hit CTRL-C to quit (the file has been saved).

Repeat the procedure for the second scattering path. Don't forget to normalize by the number of neighbors (6)! (*This last point is the most common mistake people make!!*)

2. Performing a fit

We now have a reduced data file and two standard files to fit the data, so we are finally ready for our first fit. That brings us to the program RSFIT. Although you can run the example below, you should optain the manual for this program (see http://lise.lbl.gov/RSXAPfor the latest version). A few points:

- I/O for starting parameters, final fit parameters, etc., are in ASCII-format files, so editing is easily accomplished with your favorite text editor (if you are a linux or unix newbie, try "nedit". The screenshots, etc. in this tutorial use "vi", which I can't recommend unless you want to be a unix "pro", in which case you should also consider "emacs".)
- Once you get used to the format, it is very easy and quick to edit the input files to make changes.
- rsfit will not overwrite existing files, rather, it appends a version number.
- Final fit parameters are output in two formats, one of which is a new input file with the final parameters as the new starting parameters.
- etc.

The input file must be in an "input" directory, so type

cd ../input

and create an input file called "short". The input file for fitting the Tl2Mn2O7 data with the

two Tl-O standards looks like:

```
OUTFILE short
KRANGE 2.5 15.0 0.3 3
RRANGE 1.5 4.0
MAXITER 3000
LIMITS 0.2 0.2 0.5 10 0 0
CONSTRAINT A2 A1 A1 3 0 0
CONSTRAINT E2 E1 E1 1 0 0
STDDIR tutorial
STDFILE tl_o2.14.f7 2.1415
STDFILE tl_o2.45.f7 2.4527
FITDIR tutorial
FITFILE corwin
PEAK 0.05 0.0 2.0 0 0 0
PEAK 0.05 0.0 6.0 0 0 0
STARTFIT
```

The order of these lines is not very important, but please see the RSFIT manual (rsfit_man.ps) for more information. All fitting commands have to proceed the STARTFIT command. The parameters have the following meanings:

"OUTFILE short" means put the final fit parameters and other information about the fit into a file "short" in the "../output" directory.

"KRANGE 2.5 15.0 0.3 3" gives the k-range of the fir (2.5-15 Ang^-1), the width of the transform window (0.3 Ang^-1), and the final weight (as in n in k^n chi) of the transform.

"RRANGE 1.5 4.0" gives the R-range of the fit (from 1.5 to 4 Ang).

"MAXITER 3000" is an optional line saying to stop the fit after 3000 iterations of the fit function.

"LIMITS 0.2 0.2 .5 10 0 0" is an optional line stating that the fit should not change certain parameters by more than some amount. In this case, the fit will not alter any sigma by more than 0.2 Ang, and bond length by more than 0.2 Ang, any amplitude by more than 50% of its starting value, any E0 by more than 10 eV, and the third and fourth cumulants are to remain fixed.

"CONSTRAINT A2 A1 A1 3 0 0" is an optional line constraining the amplitudes of the 1st and 2nd peaks. The syntax is Pfix P1 P2 C1 C2 C3, where Pfix=C1*P1+C2*C2 + C3, the C's are constants and the P's are parameters (for instance, A1 is the ampltude of the first peak, R3 is the r-shift of the third peak, etc. See the RSFIT manual). The example says to make the amplitude of the 2nd peak 3 times than of the first. The valid parameters are A#, S#, R#, E#, T#, F#, and D# which stand for Amplitudes, Sigmas, R-shifts, E0s, C3s (Third Cumulant), C4s (Forth Cumulant), and Dummy variables. The number associated with Dummy variables are not associated with any peak.

"STDDIR tutorial" gives the run directory where the k-space standard files are stored.

"STDFILE tl_o2.14.f7 2.1414" gives the name of the file for a standard, as well as the pair length corresponding to that standard.

"FITDIR tutorial" gives the run directory where the data is stored.

"FITFILE corwin" gives the data filename.

"PEAK 0.05 0.0 3.0 0.0 0.0 0.0" Gives the starting values of sigma, r-shift, amplitude, E0-shift, C3 and C4 for the first standard (same order as the STDFILE lines). (The starting parameters of dummy variables are given in the DUMMIES line. See the RSFIT manual.)

"STARTFIT" means all the info is input, so start the fit! Another fit can be specified immediately below such a line.

Once this file is prepared, type "rsfit -vf short". The "-v" is an option for "verbose" and just tells you how the fit is doing, and the "-f" says to output the fitted function, ala:

```
[chbooth@lise input]$ rsfit -vf short
 Will output fit files.
 Input Files from ../input/
 Output Files to ../output/
 ../input/short.1
 ../output/short
Pass    477 <err>*100=  16.40 chi2=  147.24 0.000000 0.000000
 Finished fit to corwin ,Limits hit =  0
 /home/exafs/exdata/tutorial/rs/corwin_peak.zzz
 /home/exafs/exdata/tutorial/rs/corwin_rspk.zzz
 /home/exafs/exdata/tutorial/rs/corwin_diff.zzz
```

The line beginning with "Pass" is a running update. The <err>*100 value is the mean error (times 100) between the data and the fit in the same units as the data. For instance, a "good" fit to k^3Chi data should be around 2 or 3 or better. For k*Chi data a good fit is less than 1.0. The chi2 value is a quick estimate of the statistical chi^2 assuming an error of 0.002 per data point, and can be refined later. The next number is a useful diagnostic for making sure all is well; it is the C3 parameter in the first peak in the fit, and for the fits we'll discuss here it should always be zero unless there is an error in the input file. The final parameter is the current value of the step size in the first fit parameter, namely, the sigma for the first peak. The output fit files have the general filename format as shown, with the "peak" file showing the radial distribution function, the "rspk" file as the actual fit, and the "diff" file as the difference between the data and the fit. Try plotting these files to get a feel for what the final fits look like.

OK, you have now gone through the entire process of reducing and fitting a data set! More advanced topic will be discussed (eventually) below, including error analysis, background techniques and fluorescence data analysis.

III. Backgrounds

A. Pre-edge

Obtaining pre-edge backgrounds for transmission data follows the procedure in the tutorial above. There are actually three basic types of background functions that can be used, falling into two groups:

1. "free form" backgrounds. This type of background started with a fit with a polynomial to the pre-edge, but rather than dictate the function's behavior in the post-edge region, the polynomial is mereley extrapolated above the edge. Although this is the original form of the pre-edge routine in the codes, you should have very few occasions to use a free-form background. One case would be if you want to subtract a constant value from the data set.
2. "forced" backgrounds. This type of background begins, like the free-form ones, with a polynomial fit to the pre-edge. However, the fit is not merely extrapolated to the post-edge region. Rather, it is forced to follow some functional form. For transmission data, the data are forced to follow a Victoreen formula. For fluorescence data, the pre-edge fit is forced to go to zero in the "forced" fitting region, that is, above the "begin forced fit" energy.

Some notes on these backgrounds:

1. Free form

To access a free-form background, simply don't specify the edge type in the pre-edge menu. Many choices will disappear... these are parameters that are not used by the free form background. BE WARNED! No estimate of E0 will be made with this form of the background, so when you proceed to a post-edge background (mu_0) check the search range values for E0, the values of E0, and Emin, etc.

The free form fit at this time does not allow normalization of the edge step. For EXAFS work, this is fine, but for XANES work, it is very nice to be able to do such a normalization. For XANES work, you may still do a Victoreen or a fluorescence background and normalize the edge that way.

2. Victoreen

More on this in the future...

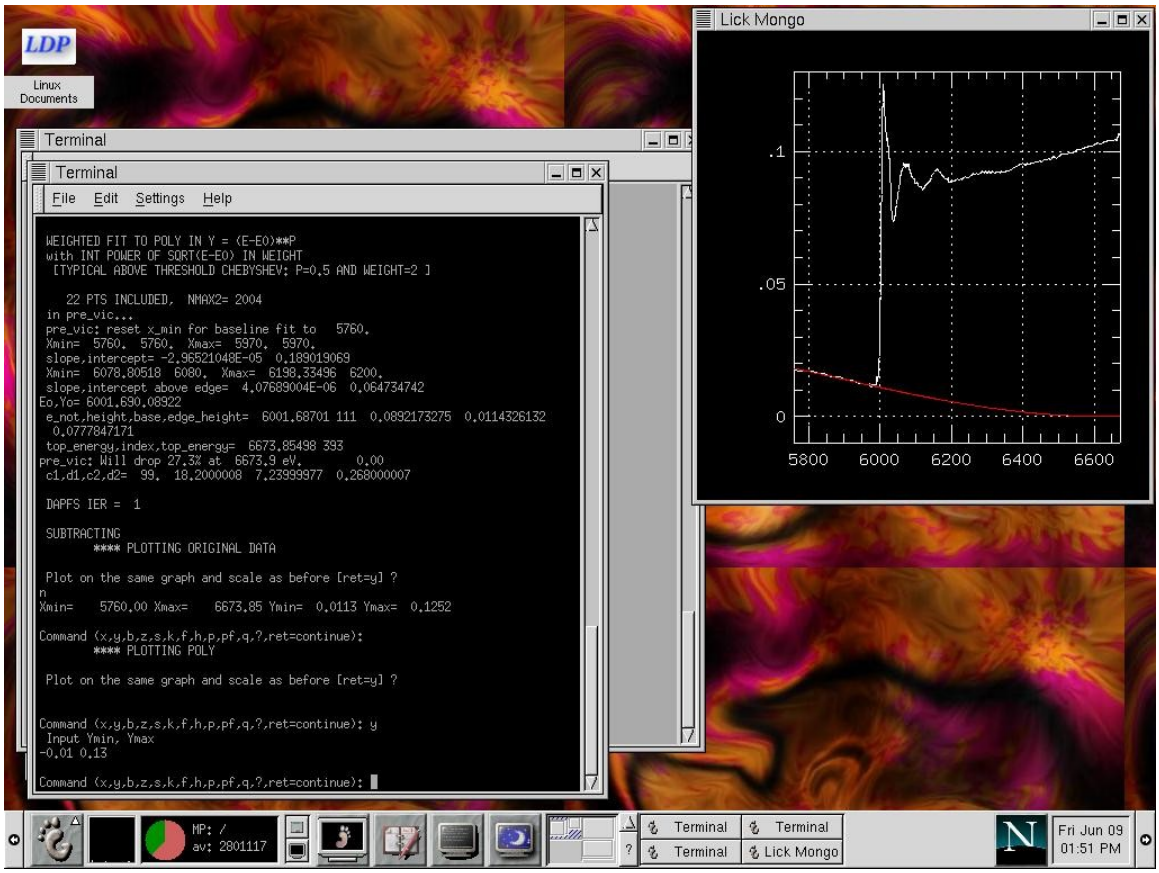
3. Fluorescence

A VERY common mistake in EXAFS analyses of fluorescence data is to remove a Victoreen-type background from the data. This is, in fact, incorrect and can actually be very different from the correct procedure. The main points here are that in fluorescence mode:

- If/I0 is NOT proportional to mu t, rather it is proportional to mu/mu0!
- IFF the desired fluorescence line is well resolved from other lines by your energy-discriminating detector, all the background absorption (fluorescence) has already been subtracted!

The first bullet basically says that if the second bullet holds true, then we should not perform ANY pre-edge subtraction of the data. However, frequently there is a DC offset to data or there is some background fluorescence. The functional form of the background fluorescence is not well understood (by me) at this time, so instead one may force a background to go to zero beyond a certain point. For instance, you may try doing a preedge subtraction of the Cr data in file "crG" in the tutorial directory:

```
**************************************************
*                                                *
*                Input file properties           *
*                                                *
**************************************************

1. Next process                    : pre-edge subtraction
2. exdata subdirectory             : may00
3. Input file                      : crG
4. Collection mode                 : fluorescence
11. Plot data file                 : n
Command (#, (h)eader, (q)uit, [continue=RET]):

 THERE ARE  393 X VALUES BETWEEN   5760.00 AND   6673.85
 *** In sbgfit***

**************************************************
*                                                *
*       Properties of PRE-edge subtraction       *
*                                                *
**************************************************
 1. Input edge [eg. U LIII, no edge=free fit]: Cr K
 2. Normalize by edge height [y,n]           : y
 3. Emin for pre-edge fit                    :     0.0 eV
 4. Emax for Pre-edge fit                    :  5970.0 eV
 5. Emin for Eo fit                          :  6080.0 eV
 6. Emax for Eo fit                          :  6200.0 eV
 7. Energy above Eo to start forced fit      :   600.0 eV
 8. End energy for forced fit (0=top)        :  6673.9 eV
 9. Power of E for Cby                       : -1.0
10. Order of Cby                             : 4
11. Plot original data [y,n]                 : y
12. Plot fit [y,n]                           : y
13. Plot subtracted data [y,n]               : y
14. Save subtracted data [y,n]               : n
Command (#, (f)resh menu, (q)uit, [continue=RET]):
```



The key parameter here is item 7, which is chosen "by eye" by roughly extrapolating where you think the background ought to go to zero.

B. Post-edge subtractions

The tutorial shows how to obtain a background using the default parameters for a fairly straight forward case. However, it can be difficult to know whether you have a "good" background, and sometimes you need to have more control. REDUCE has several features that can give you more control, and in very difficult cases, you may use REDUCE and RSFIT together to do simple fits to the residuals.

Although not discussed in the tutorial, there are two kinds of background functions you may use: splines and a chebychev polynomial.

1. Splines

Spline backgrounds are determined by choosing a number of knots, or points along the energy

axis through which a bicubic spline will be determined. The knots need not be actuall on a data point, and in fact you may place knots outside the data range. The x-postion (energy) of a knot is determined by spacing the knots equally between Emin and Emax for the fit in some power of the energy. For a power of 1, 3 knots, Emin=9000 eV and Emax=10000 eV, the knots will be placed at 9000, 9500 and 10000 eV. Emax may be given in terms of kmax. The y-positions of the knots are allowed to vary until the spline that is passed through them minimized the function $\Sigma\ y_i$-b_i, where the y_i are the data and the b_i is the polynomial background. Generally, the power of E that is chosen need not change from 1, but in some cases, it helps to have more degrees of freedom in the low energy part of the edge, in which case 0.7 may be better. In some rare cases, 0.5 may be necessary.

Some "advanced" features of the splines (accessed by choosing "a" in the Post-edge menu) include "variable" (as opposed to "fixed") knots, and choosing a range above which there will be, say x knots, and below which there will be y knots. Variable knots are chosen with the same minimization as the fixed knots, except that the x-position is also allowed to vary.

2. Chebychev polynomial

This function is useful for two reasons. First of all, its "high energy" properties are a little better than the splines: a spline knot must always be at the end of the fit range, and if another is nearby, it has a tendency to follow small oscillations at high k very well, thus reducing the measured amplitude of an EXAFS oscillation. Therefore, for disordered materials and any spectrum with little amplitude at high-k, a Chebychev can give a much more accurate background. Secondly, you may optionally specify that the fit MUST go through Emin and/or a second point (called Efix). The disadvantage to the Chebychev is that it can be difficult to get degrees of freedom in the low-E part of a background while not messing up the high-E part.

When you choose a Chebychev background(under option 10 of the post-edge menu), new choices will appear while others disappear:

```
 1. Determine Eo by linear fit (Toggle) ?    : y
 2. E0 min for linear fit                    : 12700.00
 3. E0 max for linear fit                    : 13000.00
 4. Use Kmax(K) or Emax for upper fit range  : K
 5. Kmax for post-edge fit                   : 15.00
 7. Power of (E-Eo) for post-edge fit        :  1.00
 8. Power of weight [for post-edge Cby only] :    2
10. spline or (c)heby (Toggle) ?             : C
11. Eo, the Energy (eV) for the edge         : 12666.01
12. Emin for post-edge fit                   : 12682.57
13. Order of Cby for post-edge fit           :    6
14. Plot original data?                      : n
15. Plot fit?                                : n
16. Plot results ?                           : n
17. Save results?                            : n
 Which one would you want to change?
 ([RET] next menu, (f)resh display, (a)dvanced options)
```

The main thing that appears is Option 8. A higher number here weights the high-E data more in the fit. It MUST be an INTEGER. At this point you should press "**a <RET>**" to take a look at the advanced options:

```
**************************************************
*                                                *
*            Advanced background options         *
*                                                *
**************************************************


 1. Fit for E0 interactive (toggle)          : n
 2. The second Efix for post-edge fit        : 12695.89
 3. Spline fit with fixed or var1(v) knots?  : F
 4. Using the same background as before (y)? : n
 5. Remove EXAFS from data (n)?              : n
 6. Call a standard background file (y)?     : n
 7. Use a standard background file (y)?      : n
 8. Background through points (y1,y2,y12,y21):
 9. Spline mid point energy (0=none):        :     0.00
10. # of knots below mid point:              :    0
 Which one would you want to change?
 ([RET]=last menu, (f)resh display)
```

Within this menu, you can tell REDUCE to force the polynomial through Emin and/or Efix, as well as specify Efix. The behavior is controlled through a code put into Option 8. The y's work as follows:

- y1: Force background through Emin. When doing AutoEmin, vary Emin
- y2: Force background through Efix. When doing AutoEmin, vary Efix. (WARNING: you will have to change the search range when performing AutoEmin, or strange things will happen! Fell free to try it...)
- y12: Force background through Emin AND Efix. Only vary Emin in AutoEmin.
- y21: Force background through Emin AND Efix. Only vary Efix in AutoEmin.

3. The “iterative” technique

The idea here is to take your fit to the EXAFS from RSFIT, then use it to determine the residual errors in e-space. This residual can then be used to fit a spline background which can then be directly applied to the data to regenerate the k- and r-space data. For more discussion of the applicability of this technique, please see the references in Sec. I. Here we will step through an example. Be warned! Although the concept of this procedure is childishly simple, the application is a bit convoluted.

The procedure is the following

1. Reduce the data with as good a background as you can get (within reason!!).
2. Fit the data over the range in k- and r-space that you trust

3. Generate an r-space fit file as usual EXCEPT you must extrapolate the data in k-space as far as possible (see below).
4. Use reduce to back-transform the data into k-space.
5. Generate the residuals by "Removing the EXAFS from the data"
6. Fit the residuals as if they were EXAFS
7. Use the background from step 6 to re-reduce the original, pre-edge subtracted data.
8. Go back to step 1 and repeat if necessary.

Sounds easy, doesn't it? No? Well, here is an example. It all begins with a fit... *to be continued...*

C. Difference Data

Basic procedure: It is important to have the proper backgrounds removed at the various steps. Therefore, difference data will include a version of the spectrum in the 2nd data column that has not been subtracted. If you want to generate such a data file from two separate data files (i.e. Subtract two files from each other), use the program DISH. DISH works like SHAVE... type "dish" at a command line and it will tell you what you need. The output files will have the difference of a given file with respect to the first file in a list, in column 1. Column 2 (often called the "reference" column) has the spectrum of the first file.

When reducing, you may want to subtract a function from the difference data... maybe some low frequency oscillation has crept in. When making such a subtraction with REDUCE, use the "transmission" collection mode and fit over the whole data range, rather than just the pre-edge.

You may also want to perform a pre-edge subtraction on the column-2 data. In this case, tell REDUCE the data was collected in "difference" mode. This will allow doing a pre-edge subtraction in the usual way, but column 2 will be used and *saved in column 2 of the resulting file*.

Finally, when performing the final post-edge background removal, tell REDUCE the data was collected in "difference mode" (but not to do a "pre-edge"!) and the program will take the mu_0 background from column 2 (the reference data) and apply it to the difference data in column 1, therefore normalizing the data properly.

IV. Programs list and summary

Main programs

convgerm converts SSRL ASCII data into ds and es (transmission)

sug	SuperGlitch, converts Ge-detector ds data into e-space
reduce	reduces es data into ks and rs
rsfit	performs fits in rs

Related programs from other groups

- feff7
- atoms

Utilities

plot	plots data
suplot	plots multiple data files
manip	relates to suplot
hdr	gives data files header
hedit	allows editing of binary file header (not tested)
glitch	deglitches data
sabcor	self-absorption correction separately from REDUCE
sss	prepare Z-shifted experimental standard
gete0	given a file list and a search range, finds the energy of the maximum in the 1st derivative
shave	ShiftAVErage, given a file list, will average files as well as shift the energies to match the first files reference energy
shiftES	shifts the energies of data files to a fiduciary
dish	DIfference SHave, for working with difference data
addem	OLD simple point-by-point averaging of data files
edi	OLD prog to fit and subtract one file from another
averr	Averages data files, gives stdev per point
cif2atoms	converts CIF to atoms.inp (rough)
concalc	Gives absorption for a formula and edge for samples prep
form2amu	calculates atomic mass from a formula unit
dead	analyzes dead-time scans, gives deadtime for sug
exconv	convolves a Gaussian for resolution-broadening data
kshift	performs Amp, R, E0, etc. shifts in k-space
remgaus	Narrows data in k-space
ps_shrink	shrinks a postscipt file by 50%
readxxo	translates rsfit output/ files to friendlier format
pproc	converts readxxo files to listed data files (eg. T vs. sigma2)
xxo2tex	converts output/ files to ReVTeX

Further analysis

fitcD	fit a correlated-Debye model to sigma^2 data
fitE	fit an Einstein model to sigma^2 data
getmean	given a file with x,y data with multiple measurements at each x, converts to x,y,err data
geterr	calculates stdev err from list with repeated 'x' entries
getS	strips sigma from readxxo output. Replaced by PPROC?
fites	fit e-space data
findmaxX	X=E,K,R,Real,Real0 for finding maxima (or 0) in lists of data

fitedge	fits edge data to arctans, pseudo-Voigt's, etc.
hamilton	performs Hamilton test given R(%)'s and degrees of freedom

Data conversion

convgerm	This is the main routine, handling most SSRL data (ascii only), and some APS data
aps2bin	Converts very particular GEOCARS data
renfeff	Automatically renames feffXXXX.dat files for rsfit paths
chi2bin	converts chi.dat from FEFF into RSXAP binary format
bin2gnu	converts data into ASCII (gnuplot) format
gnu2bin	converts data from ASCII (gnuplot) to RSXAP binary
k2k	input=ks file with Re and Im, bond lengths, outputs Re, amp, phase
fixendian	Performs byte swapping for porting binary data from SPARC to LINUX
fixeof	fixes possible missing final EOF at end of ASCII data

BIN2GNU, GNU2BIN

Synopsis:

Syntax: `bin2gnu` <filename, wildcards ok>
`gnu2bin` <filename, wildcards ok>

Description:

`bin2gnu` takes input files in the RSXAP binary format and converts them to a multiple column ascii format. Title, or header, lines are preceded by a '#'. The program outputs the data in a file with the same name and '.dat' appended. The order of the columns is the same as in the input file, *except* for r-space files. For instance, for an e-space file before it has been pre-edge subtracted, the ascii file will have 3 columns: energy, mu*t (sample), mu*t (reference); for k-space data, the ascii file will have 2 columns: k, k*chi. A file is determined to be in r-space merely by whether it is in a directory called "rs", and in this case the output is 4 columns: r, amplitude, -amplitude, real part. The actually binary format for r-space files (i.e., the output of bin2gnu if an r-space file is converted in a non-"rs" directory) is 3 columns: r, real part, imaginary part.

`gnu2bin` performs the inverse procedure, with the output <filename> and the input <filename.dat>

FITCD, FITD, FITE

Synopsis:

syntax: `fitcD <-we>`
`fitE`
`fitD`

options:

`-w` use 3rd datafile column for weighting data points as if the column supplied the data errors, but still assume the errors are normally distributed for the calculation of the fit parameter errors

`-e` Use the 3rd column for the statistical errors (assumes -w).

Description:

These codes fit T vs. sigma^2 data to either a correlated-Debye model (fitcD), and Einstein model (fitE) or a straight Debye model (fitD). Input instructions must be in a file called "inparam.dat", data file can have two or three columns (only two are used by default) and be in ascii. The inparam.dat file format is similar for all three programs.

fitcD

A sample input file looks like this:

```
FITFILE 14.8_S24.dat
FITRANGE 00 301
OUTFITFILE fit.dat
FITRESULTS outparams.dat
PARAMS -0.00051 163.8
VARPARAMS 1 1
MASS1 238.0
MASS2 238.0
VOLPERATOM 20.45
BONDLENGTH 4.37
```

The order of the lines is not important. Each line has a "card" followed by input data, as follows:

FITFILE *data filename*

data filename is 2 or 3 columns with the first 2 columns containing T and sigma^2.

FITRANGE *Tmin Tmax*

Range in temperature over which the fit will be calculated.

OUTFITFILE *filename*

filename will contain the actual T vs. sigma fit over the range specified by FITRANGE.

PARAMS *offset Theta_cD*

These are the starting parameters for the fit. offset is an additive constant to the calculated correlated-Debye curve (in $Å^2$), Theta_cD is the corrrelated-Debye temperature in Kelvin.

VARPARAMS *int1 int2*

int1 and int2 are integers codes indicating whether to allow the offset and/or the correlated-Debye temperature to cary in the fit. A "0" indicates to hold the parameter fixed, and a "1" indicates the parameter should be varied.

MASS1 *amu*
MASS2 *amu*

These give the mass of the absorbing and the backscattering atoms, in atomic mass units. Both are required since the model only depends on the reduced mass.

VOLPERATOM *volume*

The correlated-Debye model (unlike the Einstein and the Debye models) requires knowledge of the mean volume that an atom occupies in the system For instance, in the fcc lattice of UpdCu4, the fcc cell is 7.05 Angstroms on a side, containing 4 formula units for a mean volume of 19.47 $Angstoms^2$.

BONDLENGTH *r*

The pair distance *r* (Angstroms) for the shell in question.

<u>fitE</u>

Same as fitcD except the VOLPERATOM and BONDLENGTH cards are disabled. In the some versions of the code, these lines still must be present for the program to run, even though they are ignored. Sorry!

<u>fitD</u>

Same as fitE, except MASS2 is ignored and MASS1 is used in the model, as opposed to the reduced mass.

FITES

Synopsis:

Syntax: `fites`
Inputfile: inparam.dat
Outputfiles: various

Description:

FITES fits up to three edge files to a given edge data file, allowing for amplitude and energy shifts on each file. A sample inputfile looks like:

```
FITDIR /home/hahn/exafs/exdata/hf6/es
FITRANGE 8930 8955.
OUTFITFILE fit.dat
FITRESULTS res.dat
STDDIR /home/hahn/exafs/exdata/mag3/es
STDFILE 020b1_pre
STDFILE 020b1_pre
AMPS 1.0 .26
SHIFTS  -299.322 -306.861
VARAMPS 1 1
VARSHIFTS 1 1
FITFILE 033b1_pre
```

See the desciption of FITCD for more information, and contact me if you can't figure out any of the syntax or file formats. The number of pairs is determined by how many STDFILE lines occur in a row, and must match the number of parameters in the AMPS, SHIFTS, VARAMPS, and VARSHIFTS lines. A NEWFIT line can be appended along with a new FITFILE line. Sample output looks like:

```
 Welcome to FITES 1.05
 Fitting file
 033b1_pre
pass, lambda, 10^6*res^2=27  0.100E+03  0.161E+05
final pass: stdev(y(1))=  0.0155

 Main edge:
  4.16262865  0.0242989045 -298.691925  0.0282185636  1.12228799
0.0238809045
 -306.356232  0.098727487
 Output fit to
 fit.dat
```

The parametes listed in the "Main edge:" line are A(1),errA(1),E(1),erreE(1),A(2), etc, where A(x) is the amplitude of the xth peak, errA(x) is the error on the amplitude of the xth peak, E(x) is the energy shift to the xth peak, etc.

New parameters include RESOLUTION x.xx, where x.xx is the quadratic sum of the core-hole

lifetime and the instrumental broadening. This improves the error estimates immensely, since the number of unique data points is the total#*energy_step/resolution.

Also, a parameter called E0SAME keeps the differences between the starting E0's fixed.

Other useful flags and parameters:
STDFILEMODE transmission
FITFILEMODE transmission
STDFILEMODE reference
FITFILEMODE reference
WANTALLFITFILES
LIN

Errors are calculated from the covariance matrix of the fit, assuming the errors on the data are normally distributed.

GETE0,SHAVE

Synopsis:

Syntax: `gete0` *inputfile*
Output: deriv.bin, orig.bin, shave.lst

Syntax: `shave` *inputfile*
Output: shave.bin

Description:

These programs are designed to work together. GETE0 calculates the 1st derivative of a data set, outputting the result in a binary file "deriv.bin", together with the original data (orig.bin) scaled to fit on a plot with deriv.bin. The energy at the maximum in the derivative is recored in a file suitable for input to SHAVE. SHAVE stands for ShiftAVErage, and is used to average files together, including energy shifts if necessary.

At this time, the GETE0 input file format is rather rigid. Here is a sample:

`smooth=8.0`	This line is optional, period of smoothing, see below
`/home/exafs/exdata/may00/es`	Full path to directory where data lives
`reference`	"transmission" or "reference"
`refe0=17166.0`	Optional, indicates shifts to be relative to this energy
`104b1`	File list...
`104b2`	
`104b3`	
`104b4`	

The order of these lines is fixed. The difference between transmission and reference is really just saying "use the 1st data column" or "use the 2nd data column," respectively. This distinction is important, because a pre-edge subtracted file only has one data column, so don't use the reference line on a pre-edge subtracted file! The refe0 line is actually not used here, but is put into the output file for use by SHAVE. The ouput file "shave.lst" looks like:

```
/home/hahn/exafs/exdata/may00/es
refe0= 17166.00
104b1   17169.9512
104b2   17169.9512
104b3   17169.9512
104b4   17169.8828
```

If this is then input to SHAVE ("shave shave.lst") then each file will be shifted by roughly -3.9 eV, averaged together, and the output stored in shave.bin. If you want to use this averaged data, copy the file to the appropriate e-space directory.

The optional `smooth=`*`energy`* line indicates GETE0 to perform a Fourier transform smoothing of the data over a period of *energy* eV. Although it sounds like a lot, I have found 8 eV to

work in many cases. This option is useful for noisy data where you want to take into account that you have more data than just the two points at the peak in the derivative. For instance, a `smooth=8.0` command with 0.5 eV steps with use 16 data point to determine the derivative. However, when smoothing the data you must BE CAREFUL!!! because it is very possible the smoothing procedure will shift the measured peak away from the real peak. Generally, this shift is less than about 0.3 eV, so if you are mostly concerned with measuring relative changes between data sets and you used the same reference material, feel free about using smooth, but realize that the absolute energies may be off a bit, making absolute comparisons with other measurements more suspect. I recommend starting with a low *energy* period and observing how the measured peak derivative energies change.

Further notes about SHAVE:

SHAVE is very useful to quickly average data files even when you don't want to or can't provide energy shift information. Just don't shift the data! For example, a shave.lst file could look like:

```
/home/hahn/exafs/exdata/may00/es
104b1   0.0
104b2   0.0
104b3   0.0
104b4   0.0
```

Here the `refe0=` line is removed. When this line is not present, the shifts are calculated *relative to the first listed file*.

GLITCH

Synopsis:

Syntax: `glitch` <*inputfilename*>

Description:

GLITCH allows the user to select points in a data set and replace them with a Taylor polynomial approximation of the data; this is useful for removing monochromator glitches and sample glitches. GLITCH is designed to facilitate the deglitching process for a batch of files within a data set, particularly sets with glitches that occur repeatedly throughout the files.

Batch opening within the program:

The input file may also be specified when SUPLOT is first run, by giving the batch ("b") option; when GLITCH is started with no specified input file its output should look like the following:

```
Glich 2.02 - May 2005
Open a single file (return), or a batch (b)?
```

There is also the option to proceed by opening a single file, which is useful if only one file needs to edited, but it is not recommended that this option be used since files cannot be added at runtime, and most of the menu options behave unpredictably. Future versions of glitch will most likely remove this mode of opening files. To open a single file, I recommend using the input file method, but only listing one file in the input file.

Program operation:
Command Mode

GLITCH is based off the SUPLOT plotting scheme (see the SUPLOT manual for details), and hence navigates files in the same manner. In addition to all of the plotting commands that SUPLOT has, GLITCH has commands to enable the user to select points in a data-file for replacement by a Taylor polynomial approximation.

`m`--Enables the user to click and select multiple points from the PGPLOT window. There are several ways of operating in this mode: Using the left mouse button, the user may select two points between which GLITCH will fill in a Taylor polynomial approximation. The two points must be selected from left to right. After selecting the two points, GLITCH will request the order of the polynomial; a good rule of thumb is to give an order of n+1, where n is the number of points being replaced--though as the user you must select the point-order combination that 'best' approximates the data in that region. Also, instead of clicking the left button, the user may use the middle or right mouse buttons : the middle mouse button tells GLITCH to take the selected point and replace it with an order-2 polynomial and the right mouse button tells GLITCH to replace the selected point and the point immediately to the right with an order-3 polynomial. Also the user may associate a point-order combination with a keyboard key using

the command `type`.

`r`--pages to the next file and repeats the last glitch on it. Also, when 'repeat mode' has been toggled (by command `repeat`), **RET** has the same effect.

`type`--associates a user-specified point-order combination with a keyboard key. Once recorded, the user hits the specified keyboard key when the mouse is hovered over the desired point and the command `m` has been given.

`save`--saves the batch of files. Asks for a common suffix to append to the end of the filenames. Typically '`g`' is added to a file to denote that glitching has been performed.

HAMILTON

Synopsis:

syntax: `hamilton` [`-d`] [*Rorig*] [*DOForig*] [*Rtest*] [*DOFtest/dimension*]

Description:

This program should be run only after all fits (the original fit and the test fit) have been finalized. NOTE: There is also an option within rsfit to do Hamilton testing. Please see the RSFIT documentation for more information.

Definition of arguments:

% DEFINE THE OPTIONS - is this only the calculation you do AFTER you have
%done several rsfits? If so need to say that at beginning.

Rorig - The R value (the goodness-of-fit parameter from rsfit) for the original fit.
%For a previous fit or a fit with different
% number of parameters? Issume it is the goodness of fit parameter.
%Give some details.

DOForig - degrees of freedom for fit that yields Rorig % calculated for what situation?

Rtest - The R value for the test fit, with the peak to be tested removed.

DOFtest/dimension - Degrees of freedom for test fit or the dimension of the hypothesis (explained further in a later section).

NOTE: All arguments are optional and are defined in more detail below. If no values are given on the command line, the program enters interactive mode where one is prompted for the variables.

By default, the last argument is taken to be the degrees of freedom of the test case; If the \verb|-d| option is given, then the last argument is taken to be the dimension of the hypothesis as described below. %Define hypothesis - is it the test fit?
%or say "as described below".

Introduction:

A general problem in fitting any data set is to determine whether or not particular parameters are relevant - i.e whether including one or more of these parameters or adding an additional parameter makes a significant improvement to a given fit. Hamilton
has addressed this problem and developed a method for determining whether an increase in the number of parameters leads to a *significant* fit improvement, as defined by being some number of standard deviations away (usually three).

In fitting EXAFS data to theoretical standards the parameters come in groups associated with adding another peak in r-space; in general there are at three or more new parameter/peak, but usually some of them are constrained. Thus in EXAFS analysis, one usually would like to know if adding another peak (rather than a single parameter), -- for instance, a weak multiple scattering peak, would significantly improve the fit. Alternatively one might ask whether dropping one of the peaks included could be dropped. The addition of additional free parameters almost always improves the goodness-of-fit parameter,χ^2, giving the appearance of an improved fit. Thus, one might be tempted to continually add more and more peaks to improve the fit but one must constrain the number of free parameters to be less than the number of degrees of freedom.

Hamilton's test gives the statistical significance levels for an R-factor ratio defined by $\Re =R_1/R_0$, where R_0 and R_1 are the generalized weighted R factors for an unrestrained least-squares fit and a fit with restraints on some of the parameters, respectively

%Lisa We need to define R precisely - is it χ^2, sqrt {χ^2}, our
%value of R\%, or something else?? What do you mean by restraint? Keeping it
%constant or constraining it to another parameter, setting it zero? Need to
%clearly define this also.

Description

This program calculates the confidence level of the original fit taking into account the difference in the degrees of freedom.

% Sections to Describe/define
% 1. Paragraph or two on χ^2 and Goodness-of fit parameter. Since we
%don't use estimates of errors in our "Chi^2" what do you do instead?; Include
%an equation that defines R precisely.

% 2. Paragraph or two on how you define free parameters and Degrees of freedom and
%how you would determine them;

% 3. A Section on how constraints/restraints are included and how they might
%be developed from other structural information.

% 4. You define %\Re% in the introduction - can you explain how Hamilton
% developed this measure? and what is meant by significant changes? Also
% provide some plots to show how it varies with changes in DOF and number of
% parameters

% 5. An example of applying it to EXAFS data.`

PPROC

Synopsis:

syntax: `pproc -n=`[filenumber Default=-nothing-]

Description:

PPROC takes a RSFIT output file and extracts the position and σ^2 data from it, making plots along the way.

dependencies:
`readxxo`
`geterr`
`gnuplot`

Preparing PPROC:

pproc requires three files in the current working directory:

file 1=RSFIT output file

file 2=`inparam.dat`
`inparam.dat` is an input file that guides the action of PPROC.
It presently has three flags that it requires, and two optional.

--**required**--
`BASEFILENAME` = basename of the RSFIT output file (version number not included)
`EQUIVFILE` = a file holding the file-temperature equivilancy; see EQUIVFILE.DAT
`NUMOFSTANDARDS` = Number of Standards used in fit

--*optional*--
`POSPLOTADDENDUM` = Text to add to the end of the plot command in the position gnuplot script
`SIGPLOTADDENDUM`= Text to add to the end of the plot command in the sigma gnuplot script
`inparam.dat` must have spaces between the flags, values and the equal sign; e.g.

```
BASEFILENAME = newstd_s78
EQUIVFILE = bilayer3.dat
NUMOFSTANDARDS = 1
POSPLOTADDENDUM = "diffdata.dat" w e
```

Note: the number of standards must match the number used in the fit; output is unpredictable otherwise.

Note: PPROC places a comma after the last plot command for sigma or position if either POSPLOTADDENDUM or SIGPLOTADDENDUM flags are given; for multiple plots to be added, commas must be manually placed; e.g.

```
SIGPLOTADDENDUM = "oldsigdata.dat" w e, "oldersigdata.dat" using ($1+3):2:3 w e
```

file 3=equivfile.dat
The equivalency file is a file that tells the program what temperature corresponds to each file used the RSFIT. The file may bear any name, though the name must be specified in inparam.dat. This file follows this format:

FILENAME TEMPERATURE
FILENAME2 TEMPERATURE2
...and so on

The file must have no other text in it and no blank lines. Here is a sample:

```
092b1 3
092b2 3
092b3 3
092b4 3
093b1 50
093b2 50
093b3 50
094b1 70
094b2 70
094b3 70
095b1 100
095b2 100
095b3 100
```

Note: The routine that searches and replaces filenames with temperatures is an awk search-replace script, meaning that an entry like 092b1 3 will search out any word in the RSFIT that has 092b1 and replace it with 3. For example, this entry would replace the following with 3: 092b1G, 092b1Gg, 092b1GKg, etc.

RSFIT OUTPUT FILE--The output file from rsfit whose data you want to extract and plot

Running PPROC:
After running, pproc displays plots of the σ^2 and position data for each standard specified.
It also saves all of the data files, the gnuplot scripts used, and postscripts of the plots in a tarball (tar.gz) file that has the same name as the rsfit output file used.

Here is an example of a usage of pproc:

The following files exist in the current working directory:

inparam.dat

```
BASEFILENAME = newstd_s78
EQUIVFILE = bilayer3.dat
NUMOFSTANDARDS = 1
```

bilayer3.dat

```
092b1 3
092b2 3
092b3 3
092b4 3
093b1 50
093b2 50
093b3 50
094b1 70
094b2 70
094b3 70
095b1 100
095b2 100
095b3 100
096b1 105
096b2 105
096b3 105
097b1 112
097b2 112
097b3 112
098b1 117
098b2 117
098b3 117
099b1 130
099b2 130
099b3 130
100b1 170
100b2 170
100b3 170
101b1 200
101b2 200
102b1 200
103b1 280
103b2 280
103b3 280
```

newstd_s78.3 - the rsfit output file in use

Note that the `.3` part of the RSFIT output file was NOT specified in `inparam.dat`. This is

specified using the `-n` flag of PPROC:

```
pproc -n 3
```

This command will perform the data extraction on `newstd_s78.3`, display the data, and put all the data/plots into `newstd_s78.3.tar.gz`

SUG

Synopsis:

syntax: sug [-hv] [-d dead_sym] [-m x,y,z,...] <optional input_file>

Description:

SUG sums data from a data-space formatted file that contains fluorescence detector channels, applies dead-time corrections, allows for all or parts of channels to be excluded, and can make some attempts at automatically finding diffraction "glitches" and removing them. If no input file is given, sug goes into "interactive" mode. Otherwise it assumes "command line mode."

Command line mode

-h	prints syntax
-v	verbose output, useful when applying in non-interactive mode
-d dead_sym	apply a dead-time correction using data from dead_sym, which refers to a file in the exdata directory dead/dead.dead_sym
-m x,y,z,...	"mark" a channel or channels to not be used

In this mode, SUG looks in the current directory, so make sure you are in a "ds" directory!

Example 1:
```
may00d
sug -d LBL *b?
```

First line takes user into a "ds" directory for the run "may00", and the 2nd line converts all the files in that directory using the "LBL" detector dead-time curves. This example assumes ALL the data in the working directory is from a fluorescence detector.

Example 2:
```
may00d
sug -v -d BL112 -m 6,21,32 1[1-3][0-9]b1
```

First line takes user into a "ds" directory. The 2nd line converts only the first block of run numbers 110-139, using the BL11-2 detector file for dead-time corrections, and removing channels 6, 31, and 32 from the total sum.

Interactive mode

This mode can be activated from any directory, since it asks for the input file directory and name. SUG can plot any or all selected channels and allows for "glitching" selected regions of a given channel with a mouse. The "autoglitch" feature can detect when diffraction spikes occur in the ICR and automatically remove those data from the total sum (somewhat dangerous!). This documentation is under development, but try just typing "sug" and see how far you get... it is text-menu driven.